Quantum Monte Carlo simulation of positron annihilation radiation in solids

Jan Härkönen

School of Science

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 31.12.2019

Thesis supervisor:

Associate Prof. Lasse Leskelä

Thesis advisor:

M.Sc. Kristoffer Simula

Aalto University
School of Science

AALTO UNIVERSITY
SCHOOL OF SCIENCE

ABSTRACT OF THE
MASTER'S THESIS

Author: Jan Härkönen

Title: Quantum Monte Carlo simulation of positron annihilation radiation in solids

Date: 31.12.2019 Language: English Number of pages: 7+45+32

Department of Applied Physics, Antimatter and Nuclear Engineering

Major: Mathematics Code: SCI3054

Supervisor: Associate Prof. Lasse Leskelä

Advisor: M.Sc. Kristoffer Simula

In material studies we use a method called positron annihilation spectroscopy. It is a non-destructive method to study defects and vacancies in solids. It is based on the fact that when a positron and an electron collide, they annihilate and release electromagnetic radiation, or photons, in the form of two observable γ-quanta.

Upon annihilation, the positron-electron pair also has a momentum. The momentum distribution very much depends on the atomic structure of the sample material. In this project our focus is on the annihilation-pair momentum density (APMD), which is the probability distribution of the momentum in which the positron-electron pair annihilates in the crystal.

Instead of doing experimental measurements, we are concentrating on the simulation of a system of many electrons and one positron in a flawless periodic crystal lattice for carbon, silicon and germanium in the diamond structure. We will computationally approximate the wave function of a system, and with that, we will use the quantum Monte Carlo method to simulate the APMD and compare it with experimental results.

The main achievements of this project is creating a method to compute the APMD and gathering reference data for it with accurate simulations for well defined atomic structures. This is the first project that involves QMC simulations of non-homogeneous periodic crystal lattices.

Keywords: quantum Monte Carlo, positron, momentum density, wave function, positron annihilation, Hamiltonian, crystal lattice, Bloch theorem, simulation

AALTO-YLIOPISTO
PERUSTIETEIDEN KORKEAKOULU

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Jan Härkönen

Työn nimi: Kvantti-Monte Carlo -simulaatio positroniannihilaatiosäteilystä kiinteässä aineessa

Päivämäärä: 31.12.2019 Kieli: Englanti Sivumäärä: 7+45+32

Teknillisen Fysiikan Laitos, Antimateria ja Ydintekniikka

Pääaine: Matematiikka Koodi: SCI3054

Työn valvoja: Associate Prof. Lasse Leskelä

Työn ohjaaja: DI Kristoffer Simula

Materiaalitutkimuksessa käytetään menetelmää nimeltä positroniannihilaatiospektroskopia. Se on materiaalia vahingoittamaton menetelmä, jolla voi tutkia kiinteän aineen virheitä ja vakansseja. Se perustuu siihen, että positronin ja elektronin törmätessä ne annihiloituvat ja vapauttavat sähkömagneettista säteilyä kahden γ-kvantin muodossa, joita voi mitata.

Annihilaatiossa positroni-elektroniparilla on myös liikemäärä. Liikemääräjakauma riippuu hyvin paljon näytemateriaalin atomirakenteesta. Tämän projektin keskeinen suure on annihiloivan parin liikemäärätiheys (APMD), joka kuvaa todennäköisyysjakaumaa liikemäärälle, jolla positroni-elektronipari annihiloituu hilassa.

Sen sijaan, että teemme kokeellisia mittauksia, keskitymme simuloimaan systeemiä, jossa on useampia elektroneja ja yksi positroni virheettömässä kristallihilassa hiilelle, piille ja germaniumille timanttirakenteella. Selvitämme tietokoneella approkimaatiota systeemin aaltofunktiolle, ja sen avulla käytämme kvantti-Monte Carlo -menetelmää simuloidaksemme APMD:tä ja vertaamme sitä kokeellisiin mittaustuloksiin.

Tässä projektissa ollaan luotu tapa laskea tietokoneella APMD ja kerätty referensssidataa sille tarkoilla simulaatioilla hyvin määritetyille atomirakenteille. Tämä on ensimmäinen projekti, jossa tehdään QMC simulaatioita epähomogeenisille jaksollisille hilarakenteille.

Avainsanat: kvantti-Monte Carlo, positroni, liikemäärätiheys, aaltofunktio, positroniannihilaatio, Hamiltonin operaattori, hila, Blochin teoreema, simulaatio

Preface

I want to thank Kristoffer Simula and Ilja Makkonen for their guidance during the past 1.5 years for both my master's thesis as well as my special assignment. I'd also like to thank Filip Tuomisto and Lasse Leskelä for making this thesis possible for me.

Otaniemi, 20.11.2019

Jan Härkönen

Contents

Symbols and abbreviations

Symbols

$\mathbb{R}^n$	n-dimensional Euclidean space
$\mathbb{C}$	set of complex numbers
$\hbar$	reduced Planck's constant
e	electron charge
m_e	electron mass
ε_0	vacuum permittivity
Ψ	wave function
$\hat{\Psi}$	wave function's Fourier transform
Γ-point	the origin of the reciprocal space

Operators

$\cdot$	dot product
$\lvert\cdot\rvert$	euclidean L^2 norm of a vector
Ψ^*	complex conjugate of Ψ
$\hat{H}$	Hamiltonian operator
∇	nabla operator
∇^2	Laplacian operator
$\langle\cdot\rvert$	bra
$\lvert\cdot\rangle$	ket
$\langle\cdot\rvert\cdot\rangle$	bra-ket

Abbreviations

QMC	Quantum Monte Carlo
DFT	Density functional theory
VMC	Variational Monte Carlo
APMD	Annihilating-pair momentum density
PBC	Periodic boundary conditions
TBC	Twisted boundary conditions
ACAR	Angular correlation of annihilation radiation
BZ	Brillouin zone
a.u.	atomic units
LDA	Local density approximation
IPM	Independent particle model

1 Introduction

In this project we are studying the annihilation of electron-positron pairs, a phenomenon that can be applied in the studies of materials. The method used to study materials with positrons is called positron annihilation spectroscopy (PAS). When a positron is emitted onto a solid, it quickly thermalizes, diffuses, gets trapped and annihilates with an electron. The annihilation releases a pair of photons, or γ-rays, each travelling in opposite directions and having an energy of 511 keV [1], which can be detected with external detectors. The annihilation rate is directly related to the material's vacancy defects [2]. A vacancy is a "missing" atom in an otherwise perfect atomic lattice. The missing atom creates a potential well, where the positron can be trapped for a longer period of time before annihilation, which prolongs its lifetime, which is inversely proportional to the annihilation rate [3]. Also the lattice may have chemical defects, meaning that *e.g.* a "wrong" atom is present in lieu of a "correct" atom in the lattice. These chemical defects affect the momentum of an electron-positron pair at the annihilation point. PAS allows us to characterize the defect structure of a sample such as the character and density of defects at the atomic level, and how they affect the macroscopic properties of said material such as electrical conductivity, mechanical properties or light emission [4]. PAS is a non-destructive method, which means it doesn't damage the material in any significant way. Positron annihilation can also be applied in medicine, especially in positron emission tomography (PET) [5]. The principle is the same: we introduce a radioactive substance to the body, and the β-decay of the substance releases a positron, which annihilates and emits two detectable γ-quanta.

In this document our focus will be on the simulation aspect of PAS, by using Quantum Monte Carlo (QMC) methods [6]. We will simulate the momentum distribution in which the electron-positron pair annihilates in a perfect non-defected crystal lattice. Our chosen crystals are carbon, silicon and germanium in the diamond structure. The main physical quantity in this thesis is called annihilating-pair momentum density (APMD) [7], which quantifies the distribution of the momenta at which the electron-positron pairs annihilate. Since the APMD results of a periodic system lie on a discrete grid, we also have methods in order to get a better resolution, mainly by making a denser grid as well as shifting it in different directions. Gathering experimental data for the APMD is not exactly straightforward since there's no reference data to rely on for the experiments. By having an effective way of simulating APMD data computationally, we can predict how the APMD experimental results are expected to look like for different kinds of well defined defect structures.

Variational Monte Carlo (VMC) [6] is a QMC method for calculating high-dimensional integrals by computing the expectation value for some quantum mechanical property using random samples from an optimized trial wave function. We are going to simulate a system with many electrons and one positron using CASINO [8], a software developed in the University of Cambridge specifically for quantum Monte Carlo simulations. CASINO allows us to simulate the many-body wave function of a system, and with that we can compute the expectation value of many different properties, such as energy, electron density or in our case, momentum

density (APMD).

There is a widely used technique for computational quantum mechanic modelling called density functional theory (DFT) [4, 9, 10]. DFT is a method that lets us study electron systems only by using the electron density instead of the more complicated many-body wave function of the system. DFT doesn't describe momentum-space quantities such as the momentum density of annihilating electron-positron pairs or properly take into account electron-electron or electron-positron correlation or the correlated behaviour of particles of the exact wave function. To approximate the wave function more accurately, we are going to use the Slater-Jastrow wave function, which uses each particle's independent-electron (and independent-positron) orbitals as well as the Jastrow factor, which takes into account the correlations between particles. Monte Carlo methods are a good way for calculating accurate physical expectation values by sampling directly from the interacting many-body wave function.

The structure of this thesis is the following: first we introduce the theory and mathematics in which our project is based on. Then we introduce the methods used, and especially give more details on how QMC operates. Also we present the atomic structures used in our simulations as well as the parameters used in our software. After that we explain the steps to take in order to execute our simulations. Finally we compare our simulation results with reference data from experimental measurements documented in existing literature.

2 Defining variables

In this section we define variables, that will be used throughout this document. Let $\{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_N\}$ be the electrons' positions, where $\boldsymbol{r}_i \in \mathbb{R}^3$, $i \in \{1, \ldots, N\}$. Let $\boldsymbol{r}_0 \in \mathbb{R}^3$ be the positron's position. The reason we define only one positron position is because we will simulate a system of several electrons but only one positron, since in the experiments we only have one positron at a time in a sample. Let $\Psi(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_N)$ be the wave function of a system of N electrons and $\Psi(\boldsymbol{r}_0; \boldsymbol{r}_1, \ldots, \boldsymbol{r}_N)$ the wave function of a system of N electrons and one positron. Let

$$R' = (\boldsymbol{r}_2, \ldots, \boldsymbol{r}_N) \in \mathbb{R}^{3\times(N-1)} \tag{1}$$

$$R = (\boldsymbol{r}_1, \ldots, \boldsymbol{r}_N) \in \mathbb{R}^{3\times N} \tag{2}$$

$$R^+ = (\boldsymbol{r}_0, \ldots, \boldsymbol{r}_N) \in \mathbb{R}^{3\times(N+1)} \tag{3}$$

such that

$$\Psi(R) = \Psi(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_N) \tag{4}$$

and

$$\Psi(\boldsymbol{r}_0; \boldsymbol{r}_1, \ldots, \boldsymbol{r}_N) = \Psi(\boldsymbol{r}_0; \boldsymbol{r}_1, R') = \Psi(\boldsymbol{r}_0; R) = \Psi(R^+) = \Psi. \tag{5}$$

In quantum physics the probability density function for the positions of the particles is

$$|\Psi(R^+)|^2. \tag{6}$$

Also let

$$\mathrm{d}\boldsymbol{r}_0\,\mathrm{d}\boldsymbol{r}_1\ldots\mathrm{d}\boldsymbol{r}_N = \mathrm{d}\boldsymbol{r}_0\,\mathrm{d}\boldsymbol{r}_1\,\mathrm{d}R' = \mathrm{d}\boldsymbol{r}_0\,\mathrm{d}R = \mathrm{d}R^+. \tag{7}$$

Let the distance between particles i and j be

$$r_{ij} = |\boldsymbol{r}_i - \boldsymbol{r}_j| \tag{8}$$

and the distance between particle i and nucelus I

$$r_{iI} = |\boldsymbol{r}_i - \boldsymbol{r}_I|, \tag{9}$$

where $\boldsymbol{r}_I$ is the position of the nucleus I. Let the position of the ith particle be

$$\boldsymbol{r}_i = \begin{bmatrix} r_{ix} \\ r_{iy} \\ r_{iz} \end{bmatrix}. \tag{10}$$

Then the nabla operator for the ith particle $\nabla_i : \mathbb{R} \to \mathbb{R}^3$ is defined as

$$\nabla_i = \begin{bmatrix} \partial/\partial r_{ix} \\ \partial/\partial r_{iy} \\ \partial/\partial r_{iz} \end{bmatrix}. \tag{11}$$

Also the Laplace operator for the ith particle $\nabla_i^2 : \mathbb{R} \to \mathbb{R}$ is defined as

$$\nabla_i^2 = \frac{\partial^2}{\partial r_{ix}^2} + \frac{\partial^2}{\partial r_{iy}^2} + \frac{\partial^2}{\partial r_{iz}^2}. \tag{12}$$

Let $\boldsymbol{p} \in \mathbb{R}^3$ be the momentum of a particle such that

$$\boldsymbol{p} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}. \tag{13}$$

Let the wave function in the momentum space be $\hat{\Psi}(\boldsymbol{p})$.

3 Theory

3.1 Periodic lattice

A periodic lattice is a repeating arrangement of points. For any linearly independent basis vectors in $\mathbb{R}^3$ the subgroup of all linear combinations with integer coefficients of the basis vectors form a lattice $\mathbb{T}_p$. That means a lattice $(\mathbb{T}_p, +) \lhd (\mathbb{R}^3, +)$ is isomorphic to the additive group $(\mathbb{Z}^3, +)$. Let the three basis vectors of a lattice be $\boldsymbol{r}_{p1}, \boldsymbol{r}_{p2}$ and $\boldsymbol{r}_{p3}$. The primitive cell lattice is defined as

$$\mathbb{T}_p = \{a \cdot \boldsymbol{r}_{p1} + b \cdot \boldsymbol{r}_{p2} + c \cdot \boldsymbol{r}_{p3}\ , \quad a, b, c \in \mathbb{Z}\}. \tag{14}$$

Any vector $\boldsymbol{T} \in \mathbb{T}_p$ is called a translation vector. Because our atomic structure is periodic, then for example the external potential from the nuclei is also periodic such that at the point $\boldsymbol{r} \in \mathbb{R}^3$ the value of the potential is also the same at $\boldsymbol{r} + \boldsymbol{T}$, hence

$$V_{ext}(\boldsymbol{r}) = V_{ext}(\boldsymbol{r} + \boldsymbol{T}). \tag{15}$$

The periodicity applies also to most other quantities, such as positron or electron densities.

The primitive cell is the minimum-volume cell such that it can describe the whole space by translating it with all $\boldsymbol{T} \in \mathbb{T}_p$. It occupies a volume $\Omega \subset \mathbb{R}^3$ such that

$$\Omega = \{x \cdot \boldsymbol{r}_{p1} + y \cdot \boldsymbol{r}_{p2} + z \cdot \boldsymbol{r}_{p3}\ , \quad x, y, z \in [0\ 1)\}. \tag{16}$$

3.2 Reciprocal lattice

For a crystal lattice, the reciprocal lattice is a group $(\mathbb{G}_p, +) \lhd (\mathbb{R}^3, +)$ where any value $\boldsymbol{G} \in \mathbb{G}_p$ of the lattice represents a momentum value. This is why it is sometimes called the momentum space. The momentum space is similar to the real space lattice $\mathbb{T}_p$ such that it is an additive group isomorphic to $(\mathbb{Z}^3, +)$. We will call the reciprocal lattice the G-grid. The basis vectors of the real space as expressed in (3.1) has corresponding reciprocal lattice vectors which work as the basis vectors of the G-grid. Let them be $\boldsymbol{G}_{p1}, \boldsymbol{G}_{p2}, \boldsymbol{G}_{p3} \in \mathbb{R}^3$. Therefore the reciprocal lattice of $\mathbb{T}_p$ is

$$\mathbb{G}_p = \{a \cdot \boldsymbol{G}_{p1} + b \cdot \boldsymbol{G}_{p2} + c \cdot \boldsymbol{G}_{p3}\ , \quad a, b, c \in \mathbb{Z}\}. \tag{17}$$

The basis vectors of the reciprocal and the real-space lattice have the following relationship [11]:

$$\boldsymbol{G}_{pj} \cdot \boldsymbol{r}_{pi} = \begin{cases} 2\pi, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}. \tag{18}$$

This implies that if we use any $\boldsymbol{r} \in \mathbb{R}^3$, any $\boldsymbol{G} \in \mathbb{G}_p$ and any translation vector $\boldsymbol{T} \in \mathbb{T}_p$ as defined in (14) then

$$e^{i\boldsymbol{G}\cdot\boldsymbol{r}} = e^{i\boldsymbol{G}\cdot(\boldsymbol{r}+\boldsymbol{T})}. \tag{19}$$

The first Brillouin zone (BZ) is a subset of the reciprocal space and acts as a sort of primitive cell for the reciprocal space defined as the Wigner-Seitz cell of the reciprocal space [10]. To construct it, we start with the Γ-point (origin in $\mathbb{R}^3$ for the reciprocal space), make a set of vectors $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$ that go from the Γ-point to the closest n nuclei of the lattice. Then we make a set of planes $\{P_1, \ldots, P_n\}$ such that each plane P_i is perpendicular to $\boldsymbol{v}_i$ and such that the point $\frac{1}{2}\boldsymbol{v}_i$ is on the plane P_i for all $i \in \{1, \ldots, n\}$. The first BZ is the volume surrounding the Γ-point bounded by all of the planes $\{P_1, \ldots, P_n\}$. When the first BZ is translated for all $\boldsymbol{G} \in \mathbb{G}_p$, it covers the entire 3D space. For our simulations instead of using the first BZ we simply use points from the unit cell of the reciprocal space

$$\Omega^* = \{x \cdot \boldsymbol{G}_{p1} + y \cdot \boldsymbol{G}_{p2} + z \cdot \boldsymbol{G}_{p3}\ , \quad x, y, z \in [0, 1)\} \tag{20}$$

because it is easier to program and every point in Ω^* can be translated into the first BZ.

3.3 Bra-ket notation

The bra-ket notation is used to help describe the wave function mathematically. The wave function Ψ can be considered as a point in a function space and we can define an inner-product for that space. The inner product for any two states Ψ and φ can defined such that

$$\langle\varphi|\Psi\rangle = \int \varphi^*(\boldsymbol{r})\Psi(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}. \tag{21}$$

We define the value of the wave function Ψ at the point $\boldsymbol{r}$ as

$$\Psi(\boldsymbol{r}) := \langle\boldsymbol{r}|\Psi\rangle. \tag{22}$$

This can be thought of the superposition of all the values of the wave function Ψ being projected onto the position basis $|\boldsymbol{r}\rangle$. The term $|\boldsymbol{r}\rangle$ can be thought of as a vertical "vector" of all uncountably infinite amount of position points in $\mathbb{R}^3$. In other words a sort of superposition of all the values of $\boldsymbol{r} \in \mathbb{R}^3$. The bra-notation $\langle\boldsymbol{r}|$ can be thought of being the same as $|\boldsymbol{r}\rangle$ except being a horizontal "vector" such that it is the conjugate transpose of the ket "vector", *i.e.*,

$$\langle\boldsymbol{r}| = (|\boldsymbol{r}\rangle^T)^* = |\boldsymbol{r}\rangle^\dagger. \tag{23}$$

The position operator $\hat{\boldsymbol{r}}$ has eigenvalues in $\boldsymbol{r} \in \mathbb{R}^3$ and $|\boldsymbol{r}\rangle$ represents the state of the particle in which we know with certainty to find the particle itself at $\boldsymbol{r}$. Hence

$$\hat{\boldsymbol{r}}|\boldsymbol{r}\rangle = \boldsymbol{r}|\boldsymbol{r}\rangle. \tag{24}$$

In quantum mechanics the position basis is defined in a way that

$$\langle\boldsymbol{r}|\boldsymbol{r}'\rangle = \delta(\boldsymbol{r} - \boldsymbol{r}'). \tag{25}$$

Also the following completeness property holds:

$$\int |\boldsymbol{r}'\rangle\langle\boldsymbol{r}'|\,\mathrm{d}\boldsymbol{r}' = I, \tag{26}$$

where I is the identity operator. This must be true because for example

$$\langle\boldsymbol{r}|\Psi\rangle = \langle\boldsymbol{r}|I|\Psi\rangle = \int \langle\boldsymbol{r}|\boldsymbol{r}'\rangle\langle\boldsymbol{r}'|\Psi\rangle\,\mathrm{d}\boldsymbol{r}' = \int \delta(\boldsymbol{r} - \boldsymbol{r}')\Psi(\boldsymbol{r}')\,\mathrm{d}\boldsymbol{r}' = \Psi(\boldsymbol{r}). \tag{27}$$

3.4 Hamiltonian and Schrödinger's equation

An observable is a physical quantity that can be measured. In quantum mechanics an observable is the eigenvalue of an operator. The operator for the energy is Hamilton's operator $\hat{H}$. The Schrödinger's equation is used to find the total energy and wave function of a system in its stationary eigenstate.

$$\hat{H}|\Psi\rangle = E|\Psi\rangle. \tag{28}$$

The energy E is the eigenvalue of the Hamiltonian whereas the state $|\Psi\rangle$ is the eigenstate corresponding to the eigenvalue of E. The time-dependent Schödinger equation is

$$\hat{H}|\Psi(t)\rangle = i\hbar\frac{\mathrm{d}}{\mathrm{d}t}|\Psi(t)\rangle, \tag{29}$$

The single particle Hamiltonian consists of the sum of kinetic and potential energies:

$$\hat{H} = \hat{T} + \hat{V} = -\frac{\hbar^2}{2m}\nabla^2 + V(\boldsymbol{r}) = \frac{|\boldsymbol{p}|^2}{2m} + V(\boldsymbol{r}), \tag{30}$$

where $\boldsymbol{p}$ is the momentum of a particle, such that the momentum operator is

$$\hat{\boldsymbol{p}} = -i\hbar\nabla. \tag{31}$$

In this project we are using atomic units (a.u.), which eliminates the need for natural constants in SI units, meaning

$$\hbar = e = m_e = 4\pi\varepsilon_0 = 1. \tag{32}$$

The length unit in a.u. is called a Bohr corresponding to

$$1 \text{ bohr } = 0.529\text{Å} \tag{33}$$

and the energy unit is called a hartree (Ha), which is

$$1 \text{ Ha } = 27.2 \text{ eV}. \tag{34}$$

Also there is an energy unit Rydberg (Ry) which is double of a hartree *i.e.*,

$$1 \text{ Ha} = 2 \text{ Ry}. \tag{35}$$

The non-relativistic Hamilton's operator for a many-body system in an atomic structure of N electrons, one positron and N_n nuclei is defined as

$$\hat{H} = -\frac{1}{2}\sum_{i=0}^{N}\nabla_i^2 + \sum_{i=0}^{N}\sum_{I=1}^{N_n}\frac{Z_I q_i}{r_{iI}} + \sum_{i=0}^{N}\sum_{j>i}\frac{q_i q_j}{r_{ij}}, \tag{36}$$

where i and j indicate the indexes of the particles and I is the index for the nucleus of an atom. Recall in chapter 2 that when $i, j \in \{1, \ldots, N\}$ then we're talking about an electron, and when $i, j = 0$ it's a positron. Therefore the charges are $q_i = -1$ for an electron and $q_0 = 1$ for a positron. The distances r_{ij} and r_{iI} are the distances between particles and the distances between the nucleus and particle defined in (8) and (9). The number Z_I indicates the nuclear charge for nucleus I.

The Hamiltonian (36) is written with the Born-Oppenheimer approximation [12]. We use this approximation because we don't need to consider the kinetic energy of the nuclei. The mass of a nucleus is many orders of magnitude larger than an electron's or a positron's mass. The addition of the kinetic energy of the nuclei is

insignificant to the overall energy. The Born-Oppenheimer approximation treats the nuclei's charges as point charges. The Hamiltonian's first term is for the kinetic energies of the particles:

$$\hat{T} = -\frac{1}{2}\sum_{i=0}^{N} \nabla_i^2. \tag{37}$$

The second term is the Coulomb potential between particles and nuclei (external potential):

$$\hat{V}_{ext} = \sum_{i=0}^{N}\sum_{I=1}^{N_n} \frac{Z_I q_i}{r_{iI}}. \tag{38}$$

The third term is the Coulomb potential between the particles themselves (internal potential):

$$\hat{V}_{int} = \sum_{i=0}^{N}\sum_{j>i} \frac{q_i q_j}{r_{ij}}. \tag{39}$$

More details can be found in [13]. Using the plane wave solution to Schrödinger's equation of a single free particle we get

$$\Psi(\boldsymbol{r}, t) = Ce^{i(\boldsymbol{p}\cdot\boldsymbol{r} - Et)}, \tag{40}$$

where C is a constant. This plane wave is the eigenstate of the momentum operator $\hat{\boldsymbol{p}}$ defined in (31).

3.5 Fourier transform

Any periodic function can be expressed as a Fourier series. By using the property (19) and knowledge on Fourier theory we can deduce that for any function $f : \mathbb{R}^3 \to \mathbb{C}$ that is periodic in the lattice $\mathbb{T}_p$ we have the formula

$$f(\boldsymbol{r}) = \sum_{\boldsymbol{G}\in\mathbb{G}_p} \hat{f}(\boldsymbol{G}) \cdot e^{i\boldsymbol{G}\cdot\boldsymbol{r}}. \tag{41}$$

where $\hat{f}(\boldsymbol{G}) \in \mathbb{C}$ is the Fourier coefficient and $\mathbb{G}_p$ is the reciprocal lattice of $\mathbb{T}_p$. The Fourier transform of the wave function (or any other function) is a function of momentum $\boldsymbol{p}$ such that

$$\hat{\Psi}(\boldsymbol{p}) = \frac{1}{V}\int_{\Omega} e^{-i\boldsymbol{p}\cdot\boldsymbol{r}}\Psi(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}, \tag{42}$$

where V is the volume of the primitive cell which is the regular Lebesgue measure of Ω defined in (16) *i.e.*, $V = |\Omega|$. Note that in this case if the wave function $\Psi(\boldsymbol{r})$ is periodic in $\mathbb{T}_p$, then whenever $\hat{\Psi}(\boldsymbol{p})$ is nonzero it implies that $\boldsymbol{p}$ has to be in the G-grid, *i.e.*,

$$\hat{\Psi}(\boldsymbol{p}) \neq 0 \Rightarrow \boldsymbol{p} \in \mathbb{G}_p. \tag{43}$$

3.6 Momentum operator

The momentum operator $\hat{\boldsymbol{p}} = -i\hbar\nabla$ has the eigenvalue of the momentum for the momentum eigenstate $|\Psi\rangle$ such that

$$\hat{\boldsymbol{p}}|\Psi\rangle = -i\hbar\nabla|\Psi\rangle = -i\nabla|\Psi\rangle = \boldsymbol{p}|\Psi\rangle. \tag{44}$$

In (42) we showed that the wave function projected onto the momentum space is the Fourier Transform of the function in the position space:

$$\langle \boldsymbol{p}|\Psi\rangle = \hat{\Psi}(\boldsymbol{p}) = \frac{1}{V}\int_{\Omega} e^{-i\boldsymbol{p}\cdot\boldsymbol{r}}\Psi(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} = \int_{\Omega}\frac{e^{-i\boldsymbol{p}\cdot\boldsymbol{r}}}{V}\langle \boldsymbol{r}|\Psi\rangle\,\mathrm{d}\boldsymbol{r}. \tag{45}$$

By using the identity operator in (26) we know that

$$\langle \boldsymbol{p}|\Psi\rangle = \int_{\Omega}\langle \boldsymbol{p}|\boldsymbol{r}\rangle\langle \boldsymbol{r}|\Psi\rangle\,\mathrm{d}\boldsymbol{r}, \tag{46}$$

therefore the momentum basis projected onto the position basis is

$$\langle \boldsymbol{r}|\boldsymbol{p}\rangle = \langle \boldsymbol{p}|\boldsymbol{r}\rangle^{*} = \left(\frac{e^{-i\boldsymbol{p}\cdot\boldsymbol{r}}}{V}\right)^{*} = \frac{1}{V}e^{i\boldsymbol{p}\cdot\boldsymbol{r}}. \tag{47}$$

3.7 Bloch theorem for many body wave function

In order to construct the Slater-Jastrow wave function used in 4.5, we will be needing the single-particle orbitals. According to Bloch's theorem [11], a single-particle orbital in a periodic crystal can be written as

$$\phi_{\boldsymbol{k}_s}(\boldsymbol{r}) = e^{i\boldsymbol{k}_s\cdot\boldsymbol{r}}u_{\boldsymbol{k}_s}(\boldsymbol{r}), \tag{48}$$

where $u_{\boldsymbol{k}}(\boldsymbol{r})$ has the periodicity of the simulation cell lattice and $\boldsymbol{k}_s$ is a continuous quantum number that is periodic in the reciprocal space, so it is enough to examine $\boldsymbol{k}_s$ values within the first BZ. Hamilton's operator $\hat{H}$ has a translation symmetry. Applying this to our system of N electrons and a positron we can state

$$\hat{H}(\boldsymbol{r}_0; R) = \hat{H}(\boldsymbol{r}_0 + \boldsymbol{T}; R), \tag{49}$$

where $\boldsymbol{T}$ is any translation vector. The translation can just as well be applied to any other electron position $\boldsymbol{r}_i \in R$. This translation symmetry leads to the many-body Bloch condition

$$\Psi_{\boldsymbol{k}_s}(R^{+}) = e^{i\boldsymbol{k}_s\cdot\sum_{i=0}^{N}\boldsymbol{r}_i}U_{\boldsymbol{k}_s}(R^{+}), \tag{50}$$

where $U_{\boldsymbol{k}_s}$ has the periodicity of the simulation cell lattice for all particles [14, 15]. When $\boldsymbol{k}_s = 0$, then the wave function has the periodicity of the simulation cell. This is called the periodic boundary condition (PBC) [16]. The use of a nonzero simulation cell Bloch vector $\boldsymbol{k}_s$ is described as the application of twisted boundary conditions (TBC).

3.8 Annihilating-pair momentum density

3.8.1 Mathematics

The annihilating-pair momentum density (APMD) quantifies the distribution of the momentum of the electron-positron pair at the moment they annihilate [7]. Let's first examine the formula for the annihilation rate as a function of momentum from [17].

$$\Gamma(\boldsymbol{p}) = SN \int \left| \int \frac{e^{i\boldsymbol{p}\cdot\boldsymbol{r}_1}}{(2\pi)^{\frac{3}{2}}} \hat{O}_1^S \Psi(\boldsymbol{r}_1; \boldsymbol{r}_1, R') \, \mathrm{d}\boldsymbol{r}_1 \right|^2 \mathrm{d}R', \tag{51}$$

where SN is a constant and $\hat{O}_1^N$ is a spin projection operator to the singlet state of the (1st electron)-positron pair. This operator makes sure only the electron-positron pairs with opposite spins are considered. We can integrate this over $\boldsymbol{p}$ to get the total annihilation rate:

$$\lambda = \int \Gamma(\boldsymbol{p}) \, \mathrm{d}\boldsymbol{p}. \tag{52}$$

When (51) is normalized, we can consider it as the momentum density, or APMD, denoted with $\rho(\boldsymbol{p})$. Therefore

$$\rho(\boldsymbol{p}) = \frac{SN}{(2\pi)^{\frac{3}{2}}\lambda} \int \left| \int e^{i\boldsymbol{p}\cdot\boldsymbol{r}_1} \hat{O}_1^S \Psi(\boldsymbol{r}_1; \boldsymbol{r}_1, R') \, \mathrm{d}\boldsymbol{r}_1 \right|^2 \mathrm{d}R' \tag{53}$$

$$= C \int \left| \int e^{i\boldsymbol{p}\cdot\boldsymbol{r}_1} \hat{O}_1^S \Psi(\boldsymbol{r}_1; \boldsymbol{r}_1, R') \, \mathrm{d}\boldsymbol{r}_1 \right|^2 \mathrm{d}R', \tag{54}$$

where C is the constant $SN/(2\pi)^{\frac{3}{2}}\lambda$.

3.8.2 Experiments

There exists two different techniques for measuring projections of the 3D momentum density called the Doppler-broadening technique and the angular correlation of annihilation radiation (ACAR) [1]. The idea is based on the fact that the momentum of the electron-positron pair is transferred into the γ-rays released from the annihilation.

In the Doppler broadening technique we rely only on the momentum component $p_z \in \mathbb{R}$ of the propagation direction of the γ-rays. As mentioned before, the annihilation of an electron-positron pair at rest releases two γ-quanta each with an energy of 511 keV. If the electron-positron pair has a nonzero momentum component p_z in the propagation direction, then the detected energy experiences a Doppler shift deviating ΔE from the rest energy of 511 keV such that

$$\Delta E = \frac{1}{2} p_z c. \tag{55}$$

The second technique is the ACAR technique, which lets us find out the momentum component $p_{x,y} \in \mathbb{R}^2$ perpendicular to the propagation direction. This is based on the

fact that when the electron-positron pair annihilates with a momentum component perpendicular to the propagation direction, then the emitted γ-rays don't go exactly in opposite directions, but instead their paths deviate from the 180° angle by $\Theta_{x,y} \in \mathbb{R}^2$ (see figure 1). This angular deviation can be detected by external detectors. Then the momentum component $p_{x,y}$ can be examined with the equation

$$\Theta_{x,y} = \frac{p_{x,y}}{m_e c}. \tag{56}$$

Note that this is only an approximation since it only works with small angles of $\Theta_{x,y}$. Because the momentum $p_{x,y}$ is a projection of the total momentum onto the plane perpendicular the propagation direction, we call this the 2D-ACAR method. A scheme of different positron experiments can be seen in figure 1.

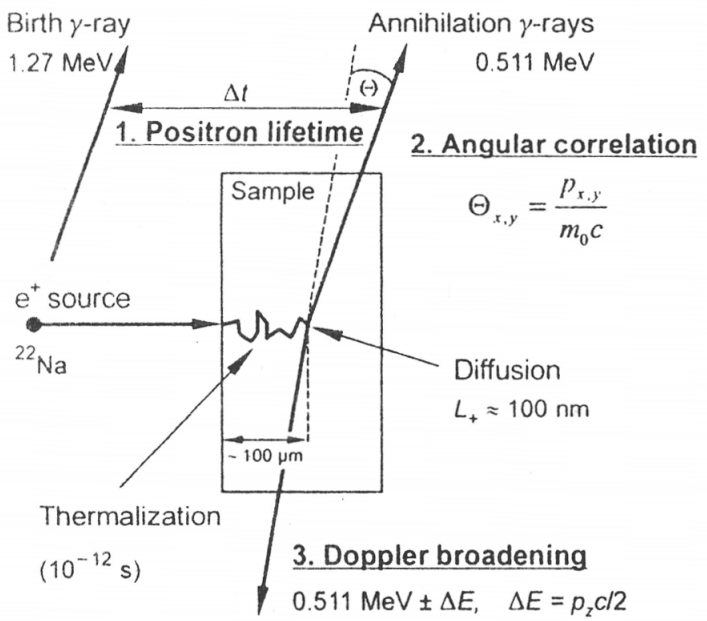

Figure 1: Scheme of positron experiments [1]

4 Methods

4.1 Pseudopotentials

A pseudopotential is an approximation of the Coulomb potential of an atomic nucleus inside a certain cutoff radius r_C. In this method we consider explicitly only the valence electrons, such that the nuclei and the core electrons are considered as "frozen" in place and are described using a pseudopotential representing the ion without the valence eletrons. A sketch of what a pseudopotential might look like can be seen in figure 2.

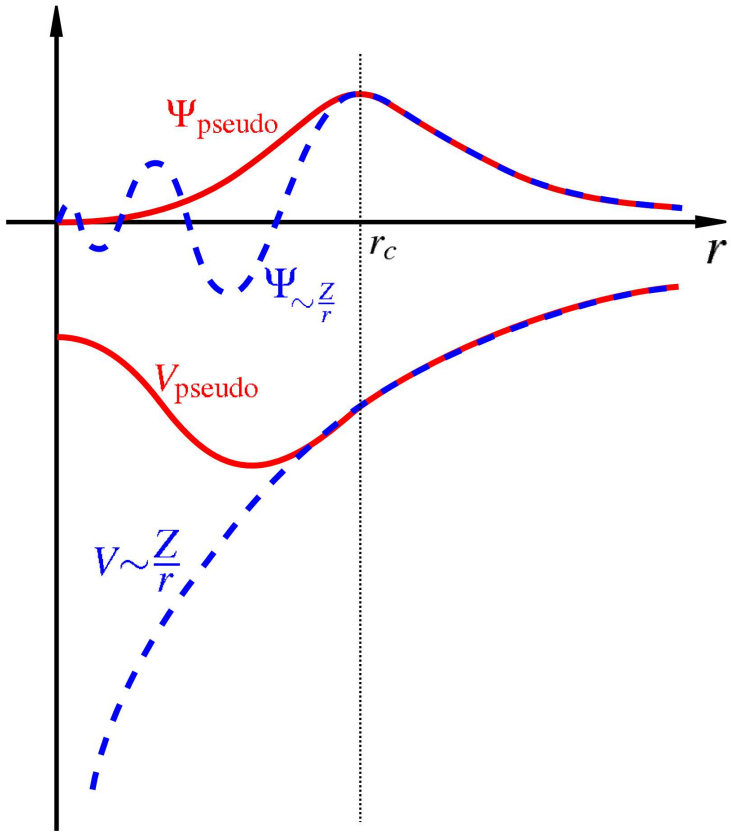

Figure 2: Comparison of a wave function Ψ using the Coulomb potential $\frac{Z}{r}$ (blue) to the one using the pseudopotential (red). See how the functions are the same beyond the core radius r_C. [18]

Basically, when approaching the nucleus, the pseudopotential is physically inaccurate, but on the other hand it doesn't diverge into values with extremely high orders of magnitude. An adequate pseudopotential should have the following 5 requirements [10, 19, 20, 21]:

(1) Real and pseudo valence eigenvalues agree for the chosen atomic reference configuration.

(2) Real and pseudo wave functions agree beyond a chosen core radius r_C.

(3) The logarithmic derivatives of the real and pseudo wave functions agree at r_C.

(4) The integrated charge from 0 to r of the real and pseudo charge densities agree for $r < r_C$ (norm conservation).

(5) The first energy derivative of the logarithmic derivatives of the real and pseudo wave functions agree at r_C and therefore for all $r \geq r_C$.

Using a pseudopotential approximation with these conditions doesn't distort the final results in any significant way. In addition to removing the tightly bound core electrons from the picture this is also useful since a suitably constructed pseudopotential enables the wave functions or orbitals to be represented using far less Fourier coefficients which makes the simulations computationally less demanding.

4.2 Quantum Monte Carlo

In quantum mechanics we want to calculate different quantum mechanical properties such as potential energy, kinetic energy, spin densities, correlations factors between particles etc. Solving a system with 1 or even 2 electrons analytically is quite straightforward. In a system of N electrons the calculations involve 3N-dimensional integrals (N particles $\times$ 3 spatial dimensions). The number for N can be small or large, but in our systems, for instance, we have $N = 64$. This involves high dimensional integrals that are downright impossible to solve analytically in a general case. Therefore we need an accurate method to approximate our integrals. For this very purpose we use Quantum Monte Carlo (QMC) methods, which approximately solves the equations of quantum mechanics which are far too complicated to solve exactly and in most cases get essentially the right answer. The principle is that we use a probability distribution to get random samples of a function. The average value of these samples is the approximation of the integral in question. The distribution itself can also be optimized for better results. This method is very efficient and the more random samples we have the better the accuracy of our results get.

4.3 Variational Monte Carlo

Variational Monte Carlo is a method based on the quantum mechanical variational principle [6], which states that for any trial wave function Ψ_T the average energy E is always at least as high as the ground-state energy E_0, *i.e.*,

$$E_0 = \frac{\langle \Psi_0 | \hat{H} | \Psi_0 \rangle}{\langle \Psi_0 | \Psi_0 \rangle} \leq \frac{\langle \Psi_T | \hat{H} | \Psi_T \rangle}{\langle \Psi_T | \Psi_T \rangle} = E, \tag{57}$$

where Ψ_0 is the ground-state wave function. The goal is to iterate the parameters of the trial wave function such that its average energy is minimized in order to construct an approximation of the ground-state wave function. As we stated in (3), the variable R^+ is an $3(N+1)$-dimensional vector containing all of the positions of the electrons and the positron. It can be randomly chosen from a probability distribution $\mathbb{P}(R^+)$, such that

$$\mathbb{P}(R^+) \geq 0, \tag{58}$$

$$\int \mathbb{P}(R^+)\, \mathrm{d}R^+ = 1. \tag{59}$$

Because the square of the absolute value of the wave function is considered to be the probability distribution of R^+ then

$$\mathbb{P}(R^+) = |\Psi_T(R^+)|^2. \tag{60}$$

Let $E_{loc}(R^+)$ be a function of the local energy such that

$$E_{loc}(R^+) = \frac{\hat{H}\Psi_T(R^+)}{\Psi_T(R^+)}. \tag{61}$$

The expectation value of the local energy is therefore

$$E = \int \mathbb{P}(R^+)E_{loc}(R^+)\,\mathrm{d}R^+ = \int |\Psi_T(R^+)|^2 E_{loc}(R^+)\,\mathrm{d}R^+. \tag{62}$$

Let's assume that Ψ_T is normalized *i.e.*, $\langle\Psi_T|\Psi_T\rangle = 1$. This means the average energy can indeed be evaluated as

$$E = \langle\hat{H}\rangle = \langle\Psi_T|\hat{H}|\Psi_T\rangle = \int \Psi_T^*(R^+)\hat{H}\Psi_T(R^+)\,\mathrm{d}R^+ \tag{63}$$

$$= \int |\Psi_T(R^+)|^2 \frac{\hat{H}\Psi_T(R^+)}{\Psi_T(R^+)}\,\mathrm{d}R^+ \tag{64}$$

$$= \int |\Psi_T(R^+)|^2 E_{loc}(R^+)\,\mathrm{d}R^+. \tag{65}$$

More details about this can be found on the papers [6] and [13].

4.4 Statistical methods

Let R_i^+ be a randomly chosen configuration from the probability distribution $|\Psi_T(R^+)|^2$. Let the expectation value and the variance of the local energy at R_i^+ be

$$\mathbb{E}(E_{loc}(R_i^+)) = E \tag{66}$$

and

$$\mathrm{Var}(E_{loc}(R_i^+)) = \sigma^2. \tag{67}$$

Therefore the average energy is the expectation value of the local energy

$$E = \langle\hat{H}\rangle = \int |\Psi_T(R^+)|^2 E_{loc}(R^+)\,\mathrm{d}R^+. \tag{68}$$

This can be evaluated with the Monte Carlo method by randomly sampling M times a configuration R_i^+ and calculating the average of every local value of the local energy:

$$E \approx E_V = \langle\hat{H}\rangle_{VMC} = \frac{1}{M}\sum_{i=1}^{M} E_{loc}(R_i^+). \tag{69}$$

This is desirable since the expectation value of the average $E_V = \langle\hat{H}\rangle_{VMC}$ is E.

Proof.

$$\mathbb{E}(E_V) = \mathbb{E}\Big(\frac{1}{M}\sum_{i=1}^{M} E_{loc}(R_i^+)\Big) \tag{70}$$

$$= \frac{1}{M}\sum_{i=1}^{M} \mathbb{E}(E_{loc}(R_i^+)) \tag{71}$$

$$= \frac{1}{M}\sum_{i=1}^{M} E = \frac{1}{M}ME = E \tag{72}$$

□

The variance of the local energy is

$$\sigma^2 = \int |\Psi_T(R^+)|^2 (E_{loc}(R^+) - E)^2 \, \mathrm{d}R^+. \tag{73}$$

Therefore the variance of the average E_V is σ^2/M.

Proof.

$$\begin{aligned}
\mathrm{Var}(E_V) &= \mathrm{Var}\Big(\frac{1}{M}\sum_{i=1}^{M} E_{loc}(R_i^+)\Big) && (74)\\
&= \frac{1}{M^2}\mathrm{Var}\Big(\sum_{i=1}^{M} E_{loc}(R_i^+)\Big) && (75)\\
&= \frac{1}{M^2}\sum_{i=1}^{M} \mathrm{Var}(E_{loc}(R_i^+)) && (76)\\
&= \frac{1}{M^2}\sum_{i=1}^{M} \sigma^2 && (77)\\
&= \frac{1}{M^2} M\sigma^2 = \frac{\sigma^2}{M} && (78)
\end{aligned}$$

□

The standard error of a variable is often considered to be the square root of the variance. Let the standard error of our average E_V be ΔE_V. When we randomly sample the local energies, we do not have the exact value for E or σ. We want to create a variable $(\Delta E_V)^2$ such that its expectation value is the variance of E_V.

$$\mathbb{E}((\Delta E_V)^2) = \mathrm{Var}(E_V) = \frac{\sigma^2}{M}. \tag{79}$$

Therefore the standard error can be computed with the formula

$$\Delta E_V = \sqrt{\frac{\langle \hat{H}^2 \rangle_{VMC} - \langle \hat{H} \rangle_{VMC}^2}{M-1}} = \sqrt{\frac{\langle \hat{H}^2 \rangle_{VMC} - E_V^2}{M-1}}. \tag{80}$$

Proof.

$$\mathbb{E}((\Delta E_V)^2) = \mathbb{E}\Big(\frac{\langle \hat{H}^2\rangle_{VMC} - E_V^2}{M-1}\Big) \tag{81}$$

$$= \frac{1}{M-1}\mathbb{E}\Big(\langle \hat{H}^2\rangle_{VMC} - E_V^2\Big) \tag{82}$$

$$= \frac{1}{M-1}\mathbb{E}\Big(\frac{1}{M}\Big[\sum_{i=1}^{M} E_{loc}(R_i^+)^2\Big] - E_V^2\Big) \tag{83}$$

$$= \frac{1}{M(M-1)}\mathbb{E}\Big(\Big[\sum_{i=1}^{M} E_{loc}(R_i^+)^2\Big] - ME_V^2\Big) \tag{84}$$

$$= \frac{1}{M(M-1)}\mathbb{E}\Big(\Big[\sum_{i=1}^{M} E_{loc}(R_i^+)^2\Big] - 2ME_V^2 + ME_V^2\Big) \tag{85}$$

$$= \frac{1}{M(M-1)}\mathbb{E}\Big(\Big[\sum_{i=1}^{M} E_{loc}(R_i^+)^2\Big] - 2ME_V\frac{1}{M}\Big[\sum_{i=1}^{M} E_{loc}(R_i^+)\Big] + ME_V^2\Big) \tag{86}$$

$$= \frac{1}{M(M-1)}\mathbb{E}\Big(\Big[\sum_{i=1}^{M} E_{loc}(R_i^+)^2\Big] - 2E_V\Big[\sum_{i=1}^{M} E_{loc}(R_i^+)\Big] + \Big[\sum_{i=1}^{M} E_V^2\Big]\Big) \tag{87}$$

$$= \frac{1}{M(M-1)}\mathbb{E}\Big(\sum_{i=1}^{M} E_{loc}(R_i^+)^2 - 2E_V E_{loc}(R_i^+) + E_V^2\Big) \tag{88}$$

$$= \frac{1}{M(M-1)}\mathbb{E}\Big(\sum_{i=1}^{M} (E_{loc}(R_i^+) - E_V)^2\Big) \tag{89}$$

$$= \frac{1}{M(M-1)}\mathbb{E}\Big(\sum_{i=1}^{M} \Big[(E_{loc}(R_i^+) - E) - (E_V - E)\Big]^2\Big) \tag{90}$$

With the same logic as using (89) to get to (84), we get

$$= \frac{1}{M(M-1)}\mathbb{E}\Big(\sum_{i=1}^{M} \Big[(E_{loc}(R_i^+) - E) - (E_V - E)\Big]^2\Big) \tag{91}$$

$$= \frac{1}{M(M-1)}\mathbb{E}\Big(\Big[\sum_{i=1}^{M} (E_{loc}(R_i^+) - E)^2\Big] - M(E_V - E)^2\Big) \tag{92}$$

$$= \frac{1}{M(M-1)}\Big[\Big[\sum_{i=1}^{M} \mathbb{E}\Big((E_{loc}(R_i^+) - E)^2\Big)\Big] - M\mathbb{E}\Big((E_V - E)^2\Big)\Big] \tag{93}$$

$$= \frac{1}{M(M-1)}\Big[\Big[\sum_{i=1}^{M} \mathrm{Var}(E_{loc}(R_i^+))\Big] - M\mathrm{Var}(E_V)\Big] \tag{94}$$

$$= \frac{1}{M(M-1)}\Big[\Big[\sum_{i=1}^{M} \sigma^2\Big] - M\frac{\sigma^2}{M}\Big] \tag{95}$$

$$= \frac{1}{M(M-1)}\Big[M\sigma^2 - \sigma^2\Big] \tag{96}$$

$$= \frac{(M-1)\sigma^2}{M(M-1)} = \frac{\sigma^2}{M}. \tag{97}$$

□

This means that the variance $(\Delta E_V)^2$ tends to zero the more configurations M we have. In the limit $M \to \infty$ we get a value that is close to E within the approximation limits of Ψ_T, and the error ΔE_V tends to zero *i.e.*, the results are more accurate the more configurations we have. It is important to note that these average values are accurate if the trial wave functions satisfy some basic conditions. Both Ψ_T and $\nabla\Psi_T$ must be continuous wherever the potential is finite, and the integrals $\int \Psi_T^* \Psi_T$ and $\int \Psi_T^* \hat{H} \Psi_T$ must exist. To keep the variance finite, $\int \Psi_T^* \hat{H}^2 \Psi_T$ also has to exist [6].

The sampling of configurations $R_i^+ \in \mathbb{R}^{3(N+1)}$ from the probability density function $|\Psi(R^+)|^2$ is not exactly straightforward. Because there is no proper inverse function to sample the configurations from, instead we use the Metropolis algorithm [6, 22]. It generates a sequence of configurations $\{R_1^+, \ldots, R_N^+\}$ by moving a so-called "walker". We start off the walker at a random position R_i^+. Then make a trial move to a new position R_{i+1}^+ according to a probability distribution $T(R_{i+1}^+ \leftarrow R_i^+)$. Then we accept the trial move with probability

$$A(R_{i+1}^+ \leftarrow R_i^+) = \min\left(1, \frac{T(R_i^+ \leftarrow R_{i+1}^+)\mathbb{P}(R_{i+1}^+)}{T(R_{i+1}^+ \leftarrow R_i^+)\mathbb{P}(R_i^+)}\right) \tag{98}$$

$$= \min\left(1, \frac{T(R_i^+ \leftarrow R_{i+1}^+)|\Psi(R_{i+1}^+)|^2}{T(R_{i+1}^+ \leftarrow R_i^+)|\Psi(R_i^+)|^2}\right). \tag{99}$$

If the trial move is accepted, the point R_{i+1}^+ becomes the next point of the walk. If not then the next point is R_i^+. The acceptance rate is desirably about 50% to minimize linear correlation. Because the wave function is constructed with the Slater determinants and the Jastrow factor (see 4.5), computing the ratio

$$\frac{\Psi(R_{i+1}^+)}{\Psi(R_i^+)} \tag{100}$$

is simple. Provided the position change applies to only one particle as a linear combination of the other particles, the ratio of the determinants is simply one coefficient of the linear combination. This makes the Metropolis algorithm very effective. The initial points generated by this algorithm depend on the starting point and are unlikely to be distributed according to our probability distribution. Eventually, however, the simulation settles down and the walkers are more distributed according to $|\Psi(R^+)|^2$. Therefore a set of initial points should be discarded from the random walk. To do this we set a number of equilibration steps in order for our distribution to settle down.

4.5 Slater-Jastrow wave function

There are various ways to approximate a wave function in CASINO. For our purposes we use the Slater-Jastrow wave function as found in [13, 23]. The approximate wave function for a system of N electrons and one positron is expressed as

$$\Psi(R^+) \approx e^{J(R^+)} D_\uparrow(R_\uparrow) D_\downarrow(R_\downarrow) \phi_0(\boldsymbol{r}_0). \tag{101}$$

The terms $D_\uparrow(R_\uparrow)$ and $D_\downarrow(R_\downarrow)$ are the Slater determinants of occupied single electron orbitals ϕ_i for particles with up and down spins respectively. The term $\phi_0(\boldsymbol{r}_0)$ is the single positron orbital. Let the number of electrons with spin-up be $N_\uparrow$ and for spin-down $N_\downarrow$, which implies that the total number of electrons is $N = N_\uparrow + N_\downarrow$. The wave functions have the antisymmetry property which means that when we switch the positions of two electrons with the same spin, then

$$\Psi(\boldsymbol{r}_0; \boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_N) = -\Psi(\boldsymbol{r}_0; \boldsymbol{r}_2, \boldsymbol{r}_1, \ldots, \boldsymbol{r}_N). \tag{102}$$

The reason Slater determinants are used is to assure that the wave function is antisymmetric. Also the fact that the $e^{J(R^+)} > 0$, and that the Jastrow factor depends only on the distances of particles, the antisymmetry holds. The Slater determinants are defined as follows:

$$D_\uparrow(R_\uparrow) = D_\uparrow(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_{N_\uparrow}) = \begin{vmatrix} \phi_1(\boldsymbol{r}_1) & \ldots & \phi_1(\boldsymbol{r}_{N_\uparrow}) \\ \vdots & \ddots & \vdots \\ \phi_{N_\uparrow}(\boldsymbol{r}_1) & \ldots & \phi_{N_\uparrow}(\boldsymbol{r}_{N_\uparrow}) \end{vmatrix} \tag{103}$$

and

$$D_\downarrow(R_\downarrow) = D_\downarrow(\boldsymbol{r}_{N_\uparrow+1}, \ldots, \boldsymbol{r}_N) = \begin{vmatrix} \phi_{N_\uparrow+1}(\boldsymbol{r}_{N_\uparrow+1}) & \ldots & \phi_{N_\uparrow+1}(\boldsymbol{r}_N) \\ \vdots & \ddots & \vdots \\ \phi_N(\boldsymbol{r}_{N_\uparrow+1}) & \ldots & \phi_{N_\uparrow}(\boldsymbol{r}_N) \end{vmatrix}, \tag{104}$$

where ϕ_i is the ith lowest independent-electron orbital. The term $J(R^+)$ is the Jastrow factor. The purpose of the Jastrow factor is to take into account the correlations between particles, which is not present in the Slater determinants alone. In our calculations we are going to use the χ, u and f terms. The Jastrow factor [24] becomes

$$J(R^+) = J(\boldsymbol{r}_0, \ldots, \boldsymbol{r}_N) = \sum_{I=1}^{N_{ions}} \sum_{i=0}^{N} \chi_I(r_{iI}) + \sum_{i<j} u(r_{ij}) + \sum_{I=1}^{N_{ions}} \sum_{i<j} f_I(r_{iI}, r_{jI}, r_{ij}), \tag{105}$$

where N_{ions} is the number of ions. The term χ characterizes the correlations between the nucleus and an electron (or positron). The term u characterizes the electron-electron (or electron-positron) correlations and the term f characterizes the electron-electron-nucleus correlations (or electron-positron-nucleus). All of these terms tend to zero, as the distances go towards infinity. They also depend on the spin and charge of the particle and a set of n free parameters, denoted with $\boldsymbol{\alpha} \in \mathbb{R}^n$, but for readability's sake the spin-factor, charge factor and $\boldsymbol{\alpha}$ have been left out of the expressions. Also the terms use a different set of parameters from $\boldsymbol{\alpha}$ whether it represents a grouping involving a positron or only electrons. The Jastrow factor can have more terms in order to get a more accurate wave function like the p and q terms, but often it is enough to use only these three terms. The p-term describes similar correlations to the u term, except that the u-term has a cutoff radius that has to fit within the simulation cell, whereas the p-term takes into account also the "corners"

of the simulation cell. The q-term on the other hand works in a similar way, except that it a describes similar correlation to the χ-term. More comprehensive details on the terms χ, u, f, p and q and about their boundary conditions can be found in [24]. Also note that when calculating the local energy with the Hamiltonian, whenever the particles coincide, the both the kinetic energy and the Coulomb potential diverge increasingly the closer their distances are from each other. These diverging values would greatly distort the approximation for any value when calculating the average value from a Monte Carlo simulation if it wasn't for application of the so-called cusp conditions. They enable the diverging kinetic and potential energies to cancel each other out by demanding that all derivatives of the wave function with respect to the particle distances have a finite eigenvalue when the distances tend to zero *i.e.*,

$$\left(\frac{\partial \Psi}{\partial r_{ij}}\right)_{r_{ij}=0} = \frac{1}{2}\Psi_{r_{ij}=0}, \tag{106}$$

when i, j are indexes for particles with opposite spins,

$$\left(\frac{\partial \Psi}{\partial r_{ij}}\right)_{r_{ij}=0} = \frac{1}{4}\Psi_{r_{ij}=0}, \tag{107}$$

when i, j are indexes for particles with the same spins and

$$\left(\frac{\partial \Psi}{\partial r_{ij}}\right)_{r_{ij}=0} = -Z\Psi_{r_{ij}=0}, \tag{108}$$

when i, j are indexes for a particle-nucleus pair with Z being the atomic number of the nucleus. In this case the local energy $\Psi^{-1}\hat{H}\Psi$ is finite even when two particles coincide. The same cusp conditions apply no matter if the particle in question is a positron or an electron [25].

The backflow function is an approximation for the wave function, that gives even more accurate results than by just using the Jastrow factor. Backflow corrections are capable of further correlations in Ψ by substituting the coordinates in the Slater determinants by a set of collective coordinates $\boldsymbol{x}_i(R^+)$, given by

$$\boldsymbol{x}_i = \boldsymbol{r}_i + \boldsymbol{\xi}_i(R^+). \tag{109}$$

Basically $\boldsymbol{x}_i$ is the particle $\boldsymbol{r}_i$ displaced by $\boldsymbol{\xi}_i$, which depends on the configuration R^+, and contains parameters that can be optimized. This method is not used for our simulations, since the optimization and simulation is too demanding when it comes to computing power and doesn't affect the results in any significant way compared to simply using the Slater-Jastrow wave function.

4.6 Optimization

The idea behind VMC optimization of the trial wave function is that the wave function has a set of free parameters $\boldsymbol{\alpha}$ in the Jastrow factor that need to be iterated for acquiring an optimized trial wave function Ψ_T. The standard optimization method

used is the variance minimization method. The idea is that any eigenstate Ψ has a constant local energy at every point R^+. This implies that the variance

$$\sigma^2 = \int |\Psi_T(R^+)|^2 |E_{loc}(R^+) - E_V|^2 \, \mathrm{d}R^+ = 0,$$

where E_V is the evaluated average energy defined in (69). In practice, however, it is impossible to simulate a flawless wave function. We can approximate the variance using VMC with

$$\sigma^2_{VMC} = \langle |\hat{H} - \langle \hat{H} \rangle_{VMC}|^2 \rangle_{VMC} > 0. \tag{110}$$

The goal is to iterate the parameters $\boldsymbol{\alpha}$ such that the variance goes as low as possible. Because Ψ_T depends on $\boldsymbol{\alpha}$, then also σ^2_{VMC} depends on $\boldsymbol{\alpha}$. We can calculate the gradient of σ^2_{VMC} with respect to the set of parameters $\boldsymbol{\alpha}$, and move the value of $\boldsymbol{\alpha}$ towards the gradient for the wave function in order to create a new trial wave function. We repeat this process until the variance of the local energy has decreased to an adequate level.

Another optimization method is the energy minimization method. In this method we use a similar method than variance minimization, but instead of minimizing the variance, we minimize the average of the energy in order get a wave function as close as possible to the ground-state energy. This method is based on the equation (57). Once we have optimized the wave function, we can evaluate any quantum mechanical property accurately such as energy, momentum density, particle density or in the case of this project the annihilating-pair momentum density (APMD) defined in 3.8.1.

4.7 Density functional theory

To construct the Slater determinants like in (103) and (104), we need to approximate the single-electron orbitals. To do this we use a method called density functional theory (DFT). It is based on the fact that the energy of the system can be expressed as a functional of the electron density. We start off with an educated guess for the electron density $n_-(\boldsymbol{r})$ such that

$$n_- : \mathbb{R}^3 \to \mathbb{R}^+. \tag{111}$$

Then we take the Kohn-Sham energy functional, which is the total amount of energy that depends on the electron density n_- [10]. It can be written in the form

$$E[n_-] = -\frac{1}{2}\sum_{i=1}^{N} \int \phi_i^*(\boldsymbol{r}) \nabla^2 \phi_i(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r} + \int n_-(\boldsymbol{r}) \hat{V}_{ext}(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r} + \frac{1}{2} \int\int \frac{n_-(\boldsymbol{r}) n_-(\boldsymbol{r}')}{|\boldsymbol{r} - \boldsymbol{r}'|} \, \mathrm{d}\boldsymbol{r} \, \mathrm{d}\boldsymbol{r}' + E_{xc}[n_-], \tag{112}$$

where $E_{xc}[n_-]$ is the exchange-correlation energy functional, which is not known exactly and has to be approximated. Also recall how $\hat{V}_{ext}$ was defined in (38).

Minimizing the total energy $E[n_-]$ leads to a set of Kohn-Sham equations for the single-electron orbitals ϕ_i:

$$\varepsilon_i\phi_i(\boldsymbol{r}) = \hat{H}_{KS}\phi_i(\boldsymbol{r}) = \Big(-\frac{1}{2}\nabla^2 + \hat{V}_{eff}(\boldsymbol{r})\Big)\phi_i(\boldsymbol{r}), \tag{113}$$

where

$$\hat{V}_{eff}(\boldsymbol{r}) = \hat{V}_{ext}(\boldsymbol{r}) + \int \frac{n_-(\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|}\,\mathrm{d}\boldsymbol{r}' + \frac{\delta E_{xc}[n_-]}{\delta n_-(\boldsymbol{r})}, \tag{114}$$

such that the last term is a functional derivative. With this we can solve the ground-state orbitals ϕ_i *i.e.*, the eigenfunctions of the Kohn-Sham Hamiltonian. The electron density is the sum of the expectation values of the position operators for each electron $\hat{n}_i(\boldsymbol{r}) = \delta(\boldsymbol{r}-\boldsymbol{r}_i)$, such that the expectation value is

$$n_-(\boldsymbol{r}) = \sum_{i=1}^{N}\langle\hat{n}_i(\boldsymbol{r})\rangle = \sum_{i=1}^{N}\langle\phi_i|\hat{n}_i(\boldsymbol{r})|\phi_i\rangle = \sum_{i=1}^{N}|\phi_i(\boldsymbol{r})|^2\,. \tag{115}$$

With this new electron density, we want to see if it is self-consistent with our initial guess. If not, then we construct a new guess from the old and new densities and repeat the process until the electron density $n_-(\boldsymbol{r})$ converges into a self-consistent value. More info in [6, 10].

4.8 Calculating the annihilating-pair momentum density with variational Monte Carlo

The main formula we're using for our simulations is found in the supplemental material of [7] in equation 8. When using configurations from the distribution $|\Psi(\boldsymbol{r}_1;\boldsymbol{r}_1,\ldots,\boldsymbol{r}_N)|^2$, the formula for the APMD is

$$\rho(\boldsymbol{G}) = \Big\langle \frac{1}{(2\pi)^3}\int \frac{\Psi(\boldsymbol{r}_0;\boldsymbol{r}_0,\boldsymbol{r}_2,\ldots,\boldsymbol{r}_N)}{\Psi(\boldsymbol{r}_1;\boldsymbol{r}_1,\boldsymbol{r}_2,\ldots,\boldsymbol{r}_N)} e^{i\boldsymbol{G}\cdot(\boldsymbol{r}_1-\boldsymbol{r}_0)}\,\mathrm{d}\boldsymbol{r}_0\Big\rangle. \tag{116}$$

By using similar notations as used previously in this document and by ignoring the normalization coefficient $1/(2\pi)^3$ then

$$\rho(\boldsymbol{G}) = \Big\langle \int \frac{\Psi(\boldsymbol{r}_0;\boldsymbol{r}_0,R')}{\Psi(\boldsymbol{r}_1;\boldsymbol{r}_1,R')} e^{i\boldsymbol{G}\cdot(\boldsymbol{r}_1-\boldsymbol{r}_0)}\,\mathrm{d}\boldsymbol{r}_0\Big\rangle. \tag{117}$$

The integral in the expectation value is estimated by Monte Carlo sampling at each configuration generated by the VMC algorithm, and the results are averaged. Next we are going to show how this relates to our APMD equation (54). Let's take this formula without the normalization coefficient C for simplicity's sake.

$$\rho(\boldsymbol{G}) = \int\Big|\int e^{i\boldsymbol{G}\cdot\boldsymbol{r}_1}\hat{O}_1^S\Psi(\boldsymbol{r}_1;\boldsymbol{r}_1,R')\,\mathrm{d}\boldsymbol{r}_1\Big|^2\,\mathrm{d}R' \tag{118}$$

The operator $\hat{O}_1^S$ in VMC is equivalent to just choosing the electron in $\boldsymbol{r}_1$ with a spin opposite to the positron's spin. Therefore we can simplify the APMD for VMC

by simply assuming that the positron in $\boldsymbol{r}_0$ and the electron in $\boldsymbol{r}_1$ have opposite spins. Hence

$$\rho(\boldsymbol{G}) = \int \left| \int e^{i\boldsymbol{G}\cdot\boldsymbol{r}_1} \Psi(\boldsymbol{r}_1; \boldsymbol{r}_1, R') \,\mathrm{d}\boldsymbol{r}_1 \right|^2 \mathrm{d}R' \tag{119}$$

$$= \int \left(\int e^{i\boldsymbol{G}\cdot\boldsymbol{r}_1} \Psi(\boldsymbol{r}_1; \boldsymbol{r}_1, R') \,\mathrm{d}\boldsymbol{r}_1 \right)^* \int e^{i\boldsymbol{G}\cdot\boldsymbol{r}_0} \Psi(\boldsymbol{r}_0; \boldsymbol{r}_0, R') \,\mathrm{d}\boldsymbol{r}_0 \,\mathrm{d}R' \tag{120}$$

$$= \int \int \int \Psi^*(\boldsymbol{r}_1; \boldsymbol{r}_1, R') e^{i\boldsymbol{G}\cdot(\boldsymbol{r}_0 - \boldsymbol{r}_1)} \Psi(\boldsymbol{r}_0; \boldsymbol{r}_0, R') \,\mathrm{d}\boldsymbol{r}_0 \,\mathrm{d}\boldsymbol{r}_1 \,\mathrm{d}R'. \tag{121}$$

Recall in (65) how the average value is related to the distribution $|\Psi|^2$ and the integral. Thus

$$\rho(\boldsymbol{G}) = \int \int \int \Psi^*(\boldsymbol{r}_1; \boldsymbol{r}_1, R') e^{i\boldsymbol{G}\cdot(\boldsymbol{r}_0 - \boldsymbol{r}_1)} \Psi(\boldsymbol{r}_0; \boldsymbol{r}_0, R') \,\mathrm{d}\boldsymbol{r}_0 \,\mathrm{d}\boldsymbol{r}_1 \,\mathrm{d}R' \tag{122}$$

$$= \int \left| \Psi(\boldsymbol{r}_1; \boldsymbol{r}_1, R') \right|^2 \int \frac{\Psi(\boldsymbol{r}_0; \boldsymbol{r}_0, R')}{\Psi(\boldsymbol{r}_1; \boldsymbol{r}_1, R')} e^{i\boldsymbol{G}\cdot(\boldsymbol{r}_0 - \boldsymbol{r}_1)} \,\mathrm{d}\boldsymbol{r}_0 \,\mathrm{d}R \tag{123}$$

$$= \left\langle \int \frac{\Psi(\boldsymbol{r}_0; \boldsymbol{r}_0, R')}{\Psi(\boldsymbol{r}_1; \boldsymbol{r}_1, R')} e^{i\boldsymbol{G}\cdot(\boldsymbol{r}_0 - \boldsymbol{r}_1)} \,\mathrm{d}\boldsymbol{r}_0 \right\rangle. \tag{124}$$

Also we can also apply the inversion symmetry of the momentum [10]. Therefore

$$\rho(\boldsymbol{G}) = \rho(-\boldsymbol{G}). \tag{125}$$

Notice that this inversion symmetry as well as the expression in (119) implies that this value has to be real. Even though the formula involves the complex number $e^{i\boldsymbol{G}\cdot(\boldsymbol{r}_1 - \boldsymbol{r}_0)}$, we expect the imaginary part of the value to converge to zero. In that case we might as well switch places for $\boldsymbol{r}_1$ and $\boldsymbol{r}_0$. Therefore

$$\rho(\boldsymbol{G}) = \left\langle \int \frac{\Psi(\boldsymbol{r}_0; \boldsymbol{r}_0, R')}{\Psi(\boldsymbol{r}_1; \boldsymbol{r}_1, R')} e^{i\boldsymbol{G}\cdot(\boldsymbol{r}_1 - \boldsymbol{r}_0)} \,\mathrm{d}\boldsymbol{r}_0 \right\rangle, \tag{126}$$

which is exactly the desired result of (117). However in a system with N electrons and one positron, the distribution used in the source code of CASINO for choosing the random configurations is $\left|\Psi(R^+)\right|^2 = |\Psi(\boldsymbol{r}_0; \boldsymbol{r}_1, R')|^2$ and not $|\Psi(\boldsymbol{r}_1; \boldsymbol{r}_1, R')|^2$ as we used above. This is why we need to adjust the formula for the VMC calculations a bit. Let's start over from (122) and apply the inversion symmetry by switch the places for $\boldsymbol{r}_1$ and $\boldsymbol{r}_0$:

$$\rho(\boldsymbol{G}) = \int \int \int \Psi^*(\boldsymbol{r}_1; \boldsymbol{r}_1, R') e^{i\boldsymbol{G}\cdot(\boldsymbol{r}_1 - \boldsymbol{r}_0)} \Psi(\boldsymbol{r}_0; \boldsymbol{r}_0, R') \,\mathrm{d}\boldsymbol{r}_0 \,\mathrm{d}\boldsymbol{r}_1 \,\mathrm{d}R' \tag{127}$$

$$= \int \int \int \left| \Psi(\boldsymbol{r}_0; \boldsymbol{r}_1, R') \right|^2 \frac{\Psi^*(\boldsymbol{r}_1; \boldsymbol{r}_1, R') \Psi(\boldsymbol{r}_0; \boldsymbol{r}_0, R')}{\left| \Psi(\boldsymbol{r}_0; \boldsymbol{r}_1, R') \right|^2} e^{i\boldsymbol{G}\cdot(\boldsymbol{r}_1 - \boldsymbol{r}_0)} \,\mathrm{d}R^+ \tag{128}$$

$$= \left\langle \frac{\Psi^*(\boldsymbol{r}_1; \boldsymbol{r}_1, R') \Psi(\boldsymbol{r}_0; \boldsymbol{r}_0, R')}{\left| \Psi(\boldsymbol{r}_0; \boldsymbol{r}_1, R') \right|^2} e^{i\boldsymbol{G}\cdot(\boldsymbol{r}_1 - \boldsymbol{r}_0)} \right\rangle. \tag{129}$$

Notice how we got rid of the integral within the average. For this average value it is apparent that we apply the position changes for only the first electron. Since all

electrons act equally with respect to the positron, it doesn't matter which electron the position changes are applied to. To save computing power, for each configuration we can apply N electron position changes to each electron at a time, and then calculate the average value. This is more efficient than sampling N different configurations. Also according to the Metropolis algorithm the position of an individual particle $\boldsymbol{r}_i$ is self-correlated, which means it depends on its own position from a previous configuration in the walker. If we computed the average by moving only the first electron like in (129) the self-correlations would be stronger and we could require more configurations in order to make the APMD values converge. By performing a position change for every electron we can diminish this correlation dramatically. Thus

$$\rho(\boldsymbol{G}) = \tag{130}$$

$$\left\langle \frac{1}{N}\sum_{i=1}^{N} \frac{\Psi^*(\boldsymbol{r}_i;\boldsymbol{r}_1,\cdots,\overbrace{\boldsymbol{r}_i}^{i:\text{th}},\cdots,\boldsymbol{r}_N)\Psi(\boldsymbol{r}_0;\boldsymbol{r}_1,\cdots,\overbrace{\boldsymbol{r}_0}^{i:\text{th}},\cdots,\boldsymbol{r}_N)}{\left|\Psi(\boldsymbol{r}_0;\boldsymbol{r}_1,R')\right|^2} e^{i\boldsymbol{G}\cdot(\boldsymbol{r}_i-\boldsymbol{r}_0)} \right\rangle. \tag{131}$$

Normally an electron-positron pair only annihilates when their spins are opposite, and this is why we needed the $\hat{O}_1^S$ operator in (51) for instance. Because in the systems considered in our simulations both the up and down-spin electrons act similarly with respect to the positron, instead of calculating over $\frac{N}{2}$ electrons, which would be the physically accurate way of doing it, we instead calculate the average over N electrons without it affecting the result of the average. This will be our main formula used to compute the APMD.

5 Increasing resolution for the results

The G-grid defined in (17) is not dense enough to get a proper graph for the APMD, which will become apparent from the results in section 7. In this section we're going to introduce methods in order to have a better resolution for the momentum density.

5.1 Supercell lattice

A supercell is when we duplicate the primitive cell arbitrarily many times in each direction. Say you have a $n \times m \times l$ supercell, where $n, m, l \in \mathbb{N}$. Let the supercell basis vectors be $\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3 \in \mathbb{R}^3$. The supercell lattice is

$$\mathbb{T}_{n\times m\times l} := \{a\cdot\boldsymbol{r}_1 + b\cdot\boldsymbol{r}_2 + c\cdot\boldsymbol{r}_3\ , \quad a,b,c\in\mathbb{Z}\} \tag{132}$$

$$= \{a\cdot n\boldsymbol{r}_{p1} + b\cdot m\boldsymbol{r}_{p2} + c\cdot l\boldsymbol{r}_{p3}\ , \quad a,b,c\in\mathbb{Z}\}. \tag{133}$$

5.2 Lattices used for the simulations

In this section we are going to introduce the atomic structure that we use for the simulations. We are going to run our simulations with 3 materials in the diamond structure: carbon, silicon and germanium (C, Si and Ge respectively). The only

difference in the structure is the distance between the atoms, which we are going to denote by a cell dimension constant a. The structure of the lattice is a periodic face-centered cubic (FCC) structure with a 2-atom basis $\mathbb{B}_p$. Let the primitive cell lattice vectors be $\boldsymbol{r}_{p1}, \boldsymbol{r}_{p2}, \boldsymbol{r}_{p3} \in \mathbb{R}^3$ which are defined as

$$\begin{bmatrix} \boldsymbol{r}_{p1} & \boldsymbol{r}_{p2} & \boldsymbol{r}_{p3} \end{bmatrix} = \frac{1}{2}a \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}. \tag{134}$$

The basis of the lattice contains 2 atoms, which are repeated with the translation vectors. The positions of the basis atoms are

$$\mathbf{0} = 0 \cdot \boldsymbol{r}_{p1} + 0 \cdot \boldsymbol{r}_{p2} + 0 \cdot \boldsymbol{r}_{p3} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{135}$$

and

$$P_a = \frac{1}{4}\boldsymbol{r}_{p1} + \frac{1}{4}\boldsymbol{r}_{p2} + \frac{1}{4}\boldsymbol{r}_{p3} = \frac{1}{4}a \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}. \tag{136}$$

Therefore the 2-atom basis $\mathbb{B}_p$ for the primitive cell is

$$\mathbb{B}_p = \{\mathbf{0} + bP_a, \quad b \in \{0, 1\}\} = \left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} a/4 \\ a/4 \\ a/4 \end{bmatrix} \right\}. \tag{137}$$

The primitive unit is illustrated in figure 3:

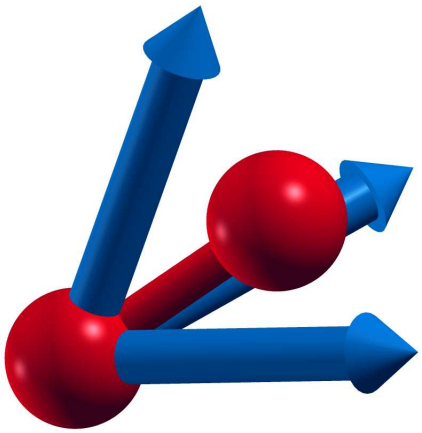

Figure 3: Primitive diamond unit with translation vectors in blue, and the nuclei in red in positions $\mathbf{0} \in \mathbb{B}_p$ and $P_a \in \mathbb{B}_p$

Similarly, the primitive diamond cell with the 2-atom basis is illustrated in figure 4:

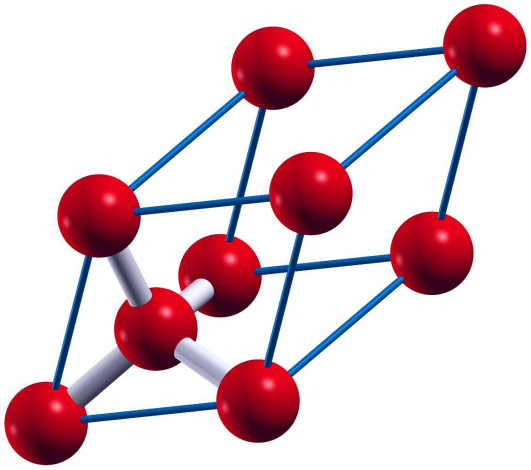

Figure 4: Primitive diamond cell

In our simulations we will use a $2 \times 2 \times 2$ supercell. Because the supercell is twice as big in every direction compared to the primitive cell, but is otherwise periodic, the lattice vectors are in the same direction as in the primitive cell, but they're twice as long. In that case our $2 \times 2 \times 2$ supercell lattice vectors are

$$\begin{bmatrix} \boldsymbol{r}_1 & \boldsymbol{r}_2 & \boldsymbol{r}_3 \end{bmatrix} = \begin{bmatrix} 2\boldsymbol{r}_{p1} & 2\boldsymbol{r}_{p2} & 2\boldsymbol{r}_{p3} \end{bmatrix} = a \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \tag{138}$$

and the lattice is of course

$$\mathbb{T} := \mathbb{T}_{2\times2\times2} = \{a \cdot \boldsymbol{r}_1 + b \cdot \boldsymbol{r}_2 + c \cdot \boldsymbol{r}_3 \ , \quad a, b, c \in \mathbb{Z}\} \tag{139}$$

$$= \{a \cdot 2\boldsymbol{r}_{p1} + b \cdot 2\boldsymbol{r}_{p2} + c \cdot 2\boldsymbol{r}_{p3} \ , \quad a, b, c \in \mathbb{Z}\}. \tag{140}$$

The supercell has a 16-atom basis $\mathbb{B}$, whose positions are

$$\mathbb{B} = \{\boldsymbol{0} + b_1\boldsymbol{r}_{p1} + b_2\boldsymbol{r}_{p2} + b_3\boldsymbol{r}_{p3} + b_4 P_a, \quad (b_1, b_2, b_3, b_4) \in \{0, 1\}^4\}. \tag{141}$$

Our $2 \times 2 \times 2$ diamond supercell is illustrated in figure 5:

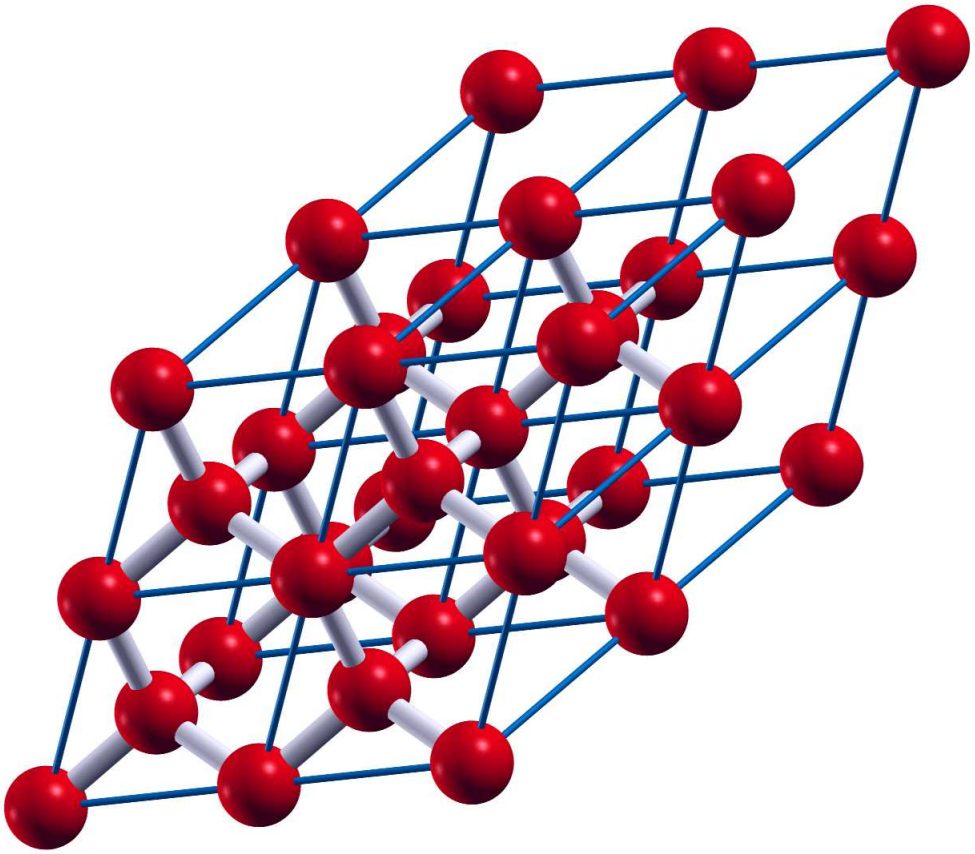

Figure 5: Diamond supercell

5.3 Increasing G-grid density

When we want to sample the APMD, we calculate the different values as a function of momentum. The values for the momentum $\boldsymbol{G}$ are determined by the G-grid $\mathbb{G}_p$. We can also make this grid twice as dense in every direction to improve the resolution. We can achieve this by instead of simulating the values using a primitive lattice $\mathbb{T}_p$, we use the $2 \times 2 \times 2$ supercell lattice $\mathbb{T}$. Obviously the relationship (18) has to hold, which implies that the supercell reciprocal lattice vectors are half as large as the primitive reciprocal lattice vectors of $\mathbb{G}_p$. Let the basis vectors for the reciprocal lattice be $\boldsymbol{G}_1, \boldsymbol{G}_2, \boldsymbol{G}_3 \in \mathbb{R}^3$. In order to satisfy the relationship (18) the basis vectors must be

$$\begin{bmatrix} \boldsymbol{G}_1 & \boldsymbol{G}_2 & \boldsymbol{G}_3 \end{bmatrix} = a^* \begin{bmatrix} 1 & -1 & 1 \\ 1 & 1 & -1 \\ -1 & 1 & 1 \end{bmatrix}, \tag{142}$$

where a^* is a constant such that $aa^* = \pi$. That makes our supercell G-grid

$$\mathbb{G} = \{a \cdot \boldsymbol{G}_1 + b \cdot \boldsymbol{G}_2 + c \cdot \boldsymbol{G}_3 \ , \quad a, b, c \in \mathbb{Z}\} \tag{143}$$

$$= \{a \cdot \frac{1}{2}\boldsymbol{G}_{p1} + b \cdot \frac{1}{2}\boldsymbol{G}_{p2} + c \cdot \frac{1}{2}\boldsymbol{G}_{p3} \ , \quad a, b, c \in \mathbb{Z}\}. \tag{144}$$

The lattice $\mathbb{G}$ is therefore reciprocal to $\mathbb{T}$ and denser than $\mathbb{G}_p$ as visualized in figure 6:

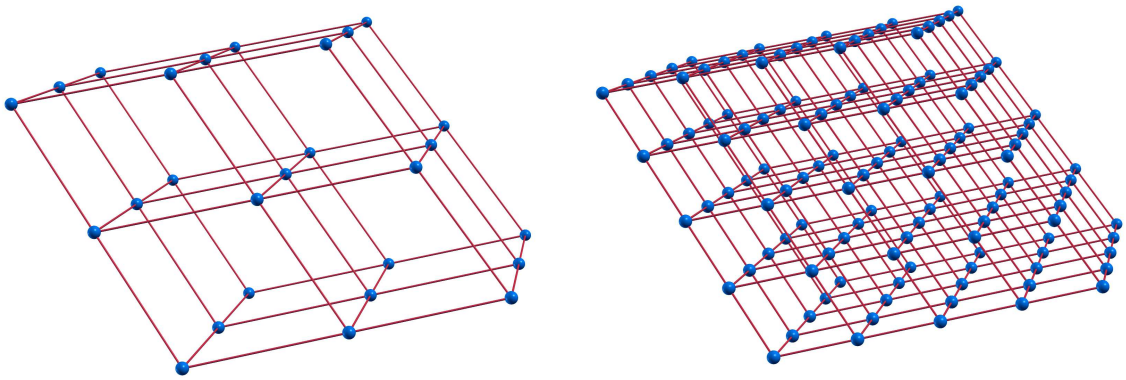

Figure 6: Primitive reciprocal lattice $\mathbb{G}_p$ (left) and supercell reciprocal lattice $\mathbb{G}$ (right).

In QMC we simulate the correlation between particles, which means that we calculate how the particles attract and repel with each other. With a supercell we get more of info on long-range correlations between the particles, instead of having the particles repeat themselves in a periodic cell that is too small. However duplicating a primitive cell too much is computationally very demanding. This is why we don't go beyond a $2 \times 2 \times 2$ supercell, but instead turn to the twist averaging method.

5.4 Twist averaging

We just made our G-grid denser, but it is not worth making it any more dense, since doing calculations for a larger supercell would require too much computing power for our purposes. In order to get more G-points, we can take the G-grid $\mathbb{G}$ and shift all of their values with an offset $\boldsymbol{k}_s$. Let's define the G-grid shifted by a $\boldsymbol{k}_s$-point:

$$\mathbb{G}_{\boldsymbol{k}_s} := \mathbb{G} + \boldsymbol{k}_s = \{a \cdot \boldsymbol{G}_1 + b \cdot \boldsymbol{G}_2 + c \cdot \boldsymbol{G}_3 \ + \boldsymbol{k}_s, \quad a, b, c \in \mathbb{Z}\}. \tag{145}$$

This implies that the $\boldsymbol{k}_s$-point is a value of momentum. The more different $\boldsymbol{k}$-points we have in different directions, the more different values we can get for the G-grid, thus getting a better resolution for our results.

Recall the how the Bloch theorem was defined in 3.7. It is important to note that when $\boldsymbol{k}_s$ is non-zero, when examining (50) the term $e^{i\boldsymbol{k}_s \cdot \boldsymbol{r}}$ does not have the periodicity of the simulation cell with respect to the electron position $\boldsymbol{r}_i$, unlike

with $U_{\boldsymbol{k}_s}$. This implies that also the wave function doesn't have the periodicity of the simulation cell. Therefore it has twisted boundary conditions (TBC) [16]. For simplicity's sake, let's work with the one particle system. By applying properties of the Fourier transform in 3.5 we know that

$$U_{\boldsymbol{k}_s}(\boldsymbol{r}) = \sum_{\boldsymbol{G}\in\mathbb{G}} \hat{U}_{\boldsymbol{k}_s}(\boldsymbol{G}) \cdot e^{i\boldsymbol{G}\cdot\boldsymbol{r}}. \tag{146}$$

Also,

$$\begin{aligned}
\hat{\Psi}_{\boldsymbol{k}_s}(\boldsymbol{p}) &= \frac{1}{V}\int_\Omega e^{-i\boldsymbol{p}\cdot\boldsymbol{r}} \cdot \Psi_{\boldsymbol{k}_s}(\boldsymbol{r}) && (147)\\
&= \frac{1}{V}\int_\Omega e^{-i\boldsymbol{p}\cdot\boldsymbol{r}} \cdot e^{i\boldsymbol{k}_s\cdot\boldsymbol{r}} U_{\boldsymbol{k}_s}(\boldsymbol{r}) && (148)\\
&= \frac{1}{V}\int_\Omega e^{-i\boldsymbol{p}\cdot\boldsymbol{r}} \cdot e^{i\boldsymbol{k}_s\cdot\boldsymbol{r}} \sum_{\boldsymbol{G}\in\mathbb{G}} \hat{U}_{\boldsymbol{k}_s}(\boldsymbol{G}) e^{i\boldsymbol{G}\cdot\boldsymbol{r}} && (149)\\
&= \frac{1}{V}\int_\Omega e^{-i\boldsymbol{p}\cdot\boldsymbol{r}} \cdot \sum_{\boldsymbol{G}\in\mathbb{G}} \hat{U}_{\boldsymbol{k}_s}(\boldsymbol{G}) e^{i(\boldsymbol{G}+\boldsymbol{k}_s)\cdot\boldsymbol{r}} && (150)\\
&= \frac{1}{V}\int_\Omega e^{-i\boldsymbol{p}\cdot\boldsymbol{r}} \cdot \sum_{\boldsymbol{G}\in\mathbb{G}_{\boldsymbol{k}_s}} \hat{U}_{\boldsymbol{k}_s}(\boldsymbol{G}-\boldsymbol{k}_s) e^{i\boldsymbol{G}\cdot\boldsymbol{r}}. && (151)
\end{aligned}$$

This implies that $\hat{\Psi}_{\boldsymbol{k}_s}(\boldsymbol{p}) = \hat{U}_{\boldsymbol{k}_s}(\boldsymbol{p}-\boldsymbol{k}_s)$. We don't go deeper into what the actual function $\hat{U}_{\boldsymbol{k}_s}$ represents, as long as we know that it depends on $\boldsymbol{p}$. The implication is that when the value $\hat{\Psi}_{\boldsymbol{k}_s}$ is non-zero then $\boldsymbol{p}$ must be in the shifted G-grid *i.e.*,

$$\hat{\Psi}_{\boldsymbol{k}_s}(\boldsymbol{p}) \neq 0 \Rightarrow \boldsymbol{p} \in \mathbb{G}_{\boldsymbol{k}_s}. \tag{152}$$

We just found a wave function, such that its Fourier transform gives us non-zero values in a shifted G-grid just by setting a suitable twisted boundary condition.

6 Simulations

6.1 Software

Quantum Espresso (QE) is one of the softwares used for our simulations [26, 27, 28]. It is an integrated suite of Open-Source computer codes for electronic-structure calculations and materials modeling at the nanoscale. Four our purposes we will be using the PWscf (Plane-Wave Self-Consistent Field) package, which is based on density-functional theory and using a plane-wave basis set and pseudopotentials. The software can be downloaded from their website [29].

The main simulation program we are using is called CASINO [8]. It is a software that has been developed in the University of Cambridge in the UK and is constantly being developed further by researchers around the globe. It was specifically developed for quantum Monte Carlo calculations. The software can be downloaded from their website [30]. CASINO is used to optimize wave functions, and to calculate expectation values for different quantum mechanical properties.

The third piece of software we use is called Atsup. It is an in-house positron DFT code, that has been under development in Otaniemi since the 80's. Atsup has an interface with CASINO created in the Master's thesis [25].

6.2 Simulation process

In this section we will go through step by step how we executed the simulations. To get started, first we have to choose the material with which we want to simulate: carbon, silicon or germanium. In this example we will use carbon (C) but every step is applicable for the other elements except for some of the file names. The contents of the files mentioned in this chapter can be seen in the attachment A. First of all we choose the ground state (GS) pseudopotential from CASINO's website [31, 32]. We save the CASINO awfn.data into the file `awfn.data` and the tabulated pseudopotenial into the file `c_pp.data`. Both of these are taken from the Dirac-Fock AREP row. Then we convert the pseudopotential into a `.UPF`-file by running the `casino2upf.x` binary file with the input `inputpp`. The command used would be

```
$QE_DIRECTORY/upftools/casino2upf.x < inputpp
```

The output `c_df.UPF` is basically a pseudopotential file in a format that can be used with PWscf. Then we will use an input file `in.pwscf` to run PWscf's `pw.x` binary file with DFT mode using the generalized gradient approximation [33]. The command is

```
$QE_DIRECTORY/bin/pw.x -pw2casino < in.pwscf > out.pwscf
```

Then we examine the total energy in the output file `out.pwscf`. We will repeat this process with several different values for the kinetic energy cutoffs in the input flag *ecutwfc* with increments of 20 Ry. The total energy converges when the cutoff energy is increased. We want to choose a cutoff as small as possible in order to make the calculations less demanding, but large enough for the energy to be close to the lowest possible energy. We choose our suitable kinetic energy cutoff by examining the difference of total energies between two increments. If the difference is less than 1 meV per basis atom, then we include the latter kinetic energy to include in the input file from this point forth. With this input file we will run the command `runpwscf` to create n Slater determinants, each having its own twist, meaning a unique value for the offset $\boldsymbol{k}_s$. The Slater determinants will be later used to construct the Slater-Jastrow wave function. An example command to make Slater determinants for 12 twists would be

```
runpwscf --qmc --twistav --xwfnstart=1 --xwfnstop=12< in.pwscf
```

The outputs are `carbon.pwfn.data.1`, ... , `carbon.pwfn.data.n`, which are Slater determinants in the plane wave basis. The way the software chooses the offset $\boldsymbol{k}_s$ for each twist is randomly generated from the first BZ. We also need to run PWscf's `pp.x` binary file with the input `in.pp` to make the file `ccharge`, which is the real space electron charge density using a Γ-centered k-mesh. We do this with the command

```
$QE_DIRECTORY/bin/pp.x <in.pp
```

We will need this file along with the plane wave files to run Atsup, which uses DFT to approximate the positron density and the ground-state positron orbital [34], which is added at the end of the `carbon.pwfn.data.n` file in order to later create a trial wave function with the positron included in the system. It can be run with the command

```
python3 $ATSUP_DIRECTORY/run_atsup.py --no_checks $1
```

The output file generated are `pwfn.data.1`, ... , `pwfn.data.n`, which now contain the Γ-point positron orbital in all of the twists. Next we migrate these plane wave files to CASINO and convert them to blip basis. The blip basis for a wave function takes much more memory than the plane-wave basis. However, since it has localized basis functions it makes it much faster to calculate the values of the wave function [35]. The blip conversion is executed with the command `blip` in the CASINO distribution. The command will ask for 4 inputs. The first one is blip multiplicity, which indicates the fineness of the blip grid. It is an integer, usually 1, but if we want a more accurate blip file then we can try with 2 or 3 for the values of the blip multiplicity. We will choose to skip the next 3 requests that are prompted. This creates files `bwfn.data.1`, ... , `bwfn.data.n`, which are Slater determinants and the positron orbital in the blip basis. For each twist we run QMC with the pseudopotential file `c_pp.data` that we used prior in PWscf. Also we need `correlation.data`, which contains the parameters $\boldsymbol{\alpha}$ for the Jastrow factor in the wave function, that have already been optimized. The optimization of the parameters is actually the most computationally demanding aspect of this whole project. The same Γ-point `correlation.data` file is used no matter what the $\boldsymbol{k}_s$-shift (aka. twist) is used. Also we need the input file `input`, which includes all the parameters included for the simulation, the most notable input parameters being the number of configurations M and equilibration steps. The command `runqmc` will execute the simulation. This creates files `expval.data.1`, ... ,`expval.data.n`, which includes a list of all the momentum values $\boldsymbol{G}+\boldsymbol{k}_s$, such that $\boldsymbol{G} \in \mathbb{G}$ and $\boldsymbol{k}_s$ is the offset. Under that there's a list of the sum of the local APMD values corresponding to the list of momentum values in their respective order:

$$\rho_{sum}(\boldsymbol{G}+\boldsymbol{k}_s) = \sum_{i=1}^{M} \rho_{loc}(\boldsymbol{G}+\boldsymbol{k}_s, R_i^+), \tag{153}$$

where R_i^+ is a configuration of particle positions chosen randomly from the distribution $|\Psi_T(R^+)|^2$. The `expval.data.n` files also include the sum of the squares of the local APMD values:

$$\rho_{square\_sum}(\boldsymbol{G}+\boldsymbol{k}_s) = \sum_{i=1}^{M} \rho_{loc}(\boldsymbol{G}+\boldsymbol{k}_s, R_i^+)^2, \tag{154}$$

where the local value of the APMD is a sample of the average in (131), except that it is not divided by N, meaning that

$$\rho_{loc}(\boldsymbol{G}+\boldsymbol{k}_s, R_i^+) = \\ \sum_{i=1}^{N} \frac{\Psi^*(\boldsymbol{r}_i; \boldsymbol{r}_1, \cdots, \boldsymbol{r}_i, \cdots, \boldsymbol{r}_N)\Psi(\boldsymbol{r}_0; \boldsymbol{r}_1, \cdots, \boldsymbol{r}_0, \cdots, \boldsymbol{r}_N)}{\left|\Psi(\boldsymbol{r}_0; \boldsymbol{r}_1, R')\right|^2} e^{i(\boldsymbol{G}+\boldsymbol{k}_s)\cdot(\boldsymbol{r}_i-\boldsymbol{r}_0)}. \tag{155}$$

We use Matlab to create the plots. To get an approximation for the APMD, we need to divide (153) with $M \cdot N$, and in order to get the sample error, we use both (153) and (154) then apply the formula (80) to get the sample error. Then all we need is to plot the data onto a graph and we're done.

In the flowchart 7, there is a summary of the steps taken with the names of the input and output files. The details on what each file is supposed to look like is in the attachment A:

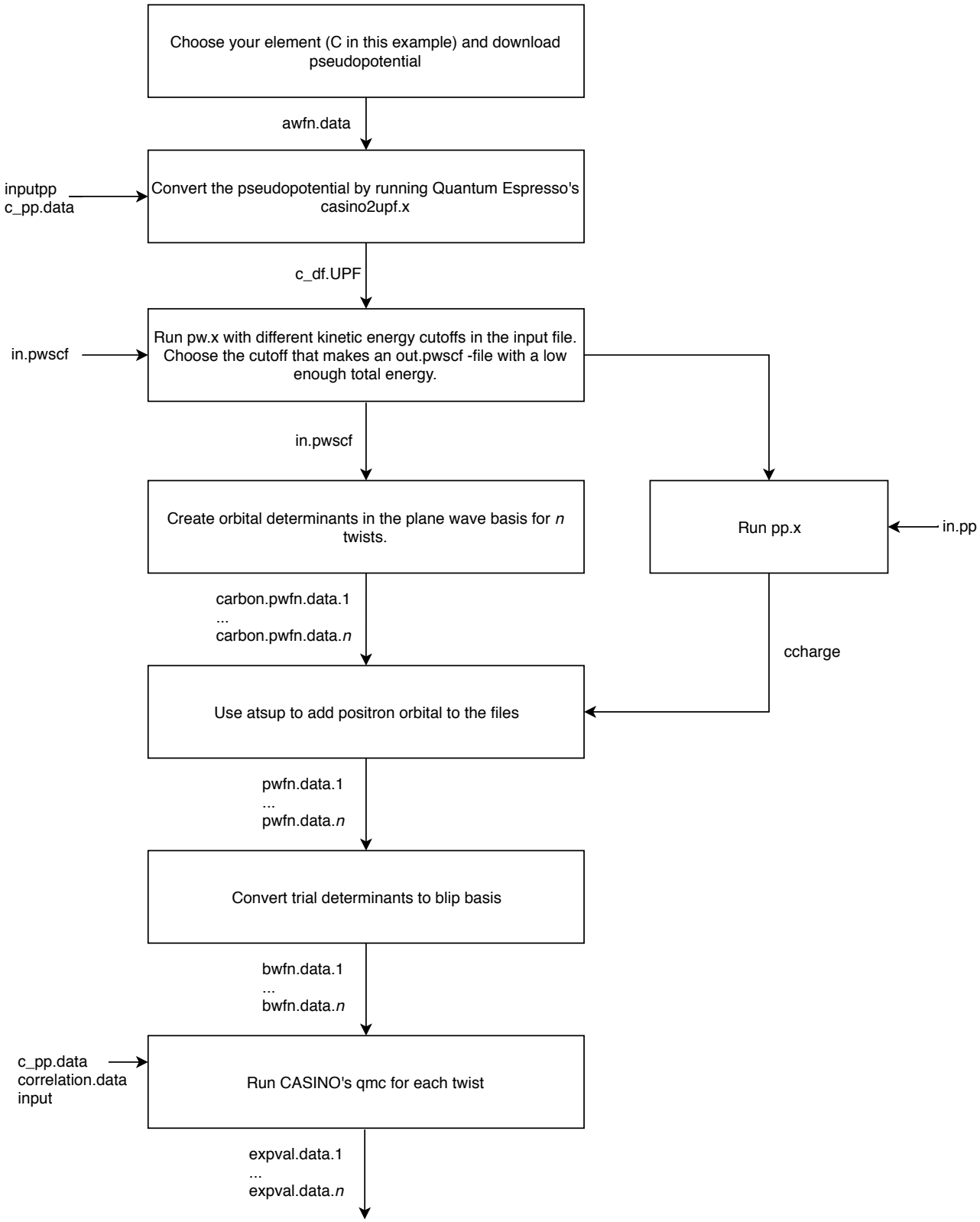

Figure 7: A flowchart of how to execute our simulations step by step.

6.3 Parameters

The constant a used in (138) for each material (diamond, silicon and germanium) are

$$a_C = 6.740652601 \text{ bohr} \tag{156}$$
$$a_{Si} = 10.261212 \text{ bohr} \tag{157}$$
$$a_{Ge} = 10.692070 \text{ bohr} \tag{158}$$

respectively. Recall how the unit of length bohr was defined in (33). For the simulations in CASINO there are quite a few key variables that are worth mentioning: the PWscf kinetic energy cutoff, the number of configurations for CASINO, number of twists, the blip multiplicity of the blip file and the number of equilibration steps for the Metropolis algorithm. Here is a table of the values we use for each parameter.

	C	Si	Ge
PWscf kinetic energy cutoff (Ry)	220	120	220
Twists	99	99	99
Blip multiplicity	3	2	1
Num. of configurations	960000	960000	960000
Equil. steps	20000	20000	20000

The reason we used a smaller blip multiplicity for the heavier elements is because we have empirically discovered, that smaller blip multiplicities don't affect the results in any significant way. It is therefore more desirable to use a lower blip multiplicity since it takes less time to generate the blip files that way. Finally our simulated system consists of $N = 64$ electrons and one positron in the simulation cell.

6.4 Reference data

The experimental reference data we are using come from [36]. Measurements have been made using the ACAR technique (see 3.8.2) on germanium and carbon. The momentum resolution of the experimental apparatus is 1.5 mrad full width at half maximum [37]. This is equivalent to the momentum value of 0.206 (a.u.) by multiplying the angular deviation of 1.5 mrad with the γ-ray's momentum. The 3D momentum densities were obtained by reconstruction using sets of 2D ACAR distributions obtained at increments of 5 degrees. Then from the 3D momentum density data, 1D plots have been made along the three high symmetry directions. The directions are expressed using regular cartesian coordinates. For this document we have digitized the data and visualized it in figure 8:

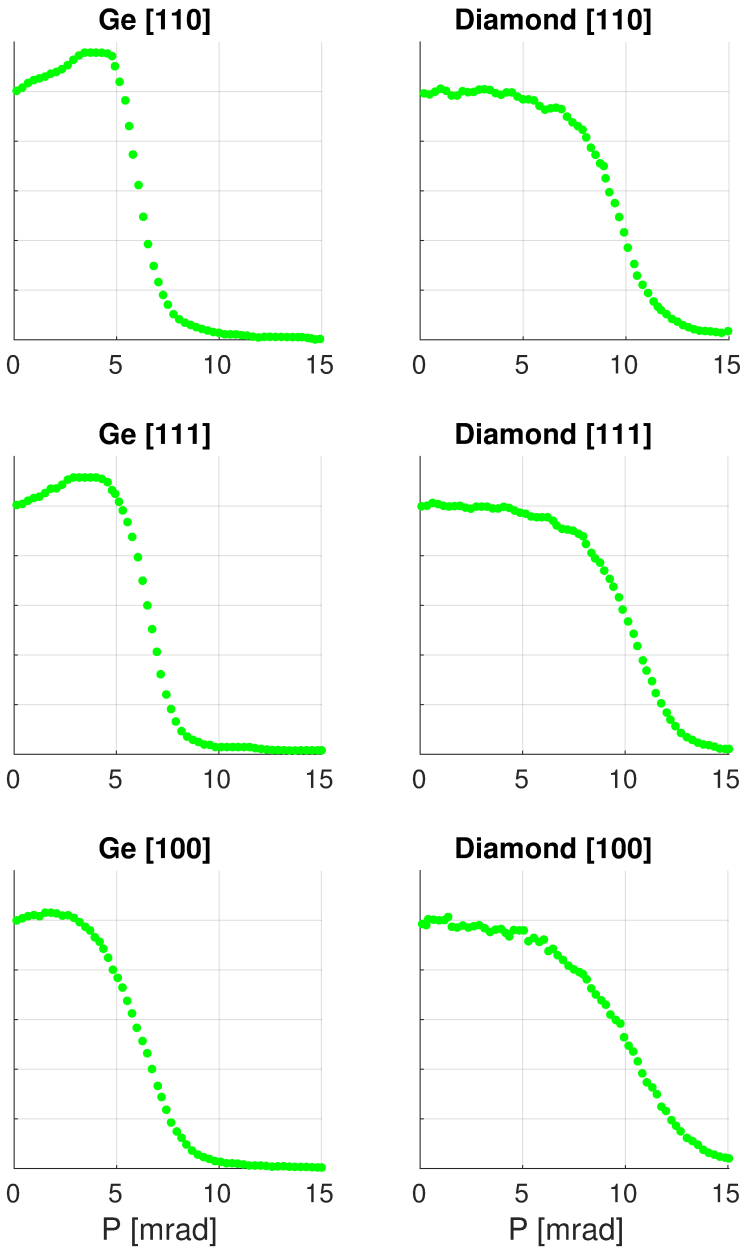

Figure 8: Experimental reference data for Ge and diamond along high symmetry directions in arbitrary units.

The x-axis is in milliradians because it was calculated with the angular deviation of the collinearity of the annihilation γ-rays, which is proportional to the momentum with small angles as stated in (56). Therefore the angular deviation is analogous with the momentum. The y-axis in the reference was expressed in arbitrary units. That doesn't matter much, since it only needs a normalization coefficient to be a proper density function.

7 Results

7.1 Monte Carlo convergence

In order to deem our Monte Carlo simulation results reliable, we need to show that as we increase the amount of data points used for the Monte Carlo simulations, the mean of the results converge to a fixed value as well as the error margins tend to zero. Recall how the mean of the positron momentum density $\rho(\boldsymbol{G})$ was calculated in (131). In this case $N = 64$, since we had a system of 64 electrons. Therefore the total the number of data points over which we want to calculate becomes

$$number\ of\ data\ points = number\ of\ configurations \cdot \underbrace{number\ of\ electrons}_{=64} = M \cdot N. \tag{159}$$

Let's take the example of $\rho(\boldsymbol{G})$ in the case where

$$\boldsymbol{G} = \begin{bmatrix} 0.932 \\ 0.932 \\ 0.000 \end{bmatrix} \tag{160}$$

and examine how the mean and the standard error act as a function of data points for a carbon lattice. Let's visualize both the real and imaginary part in figure 9:

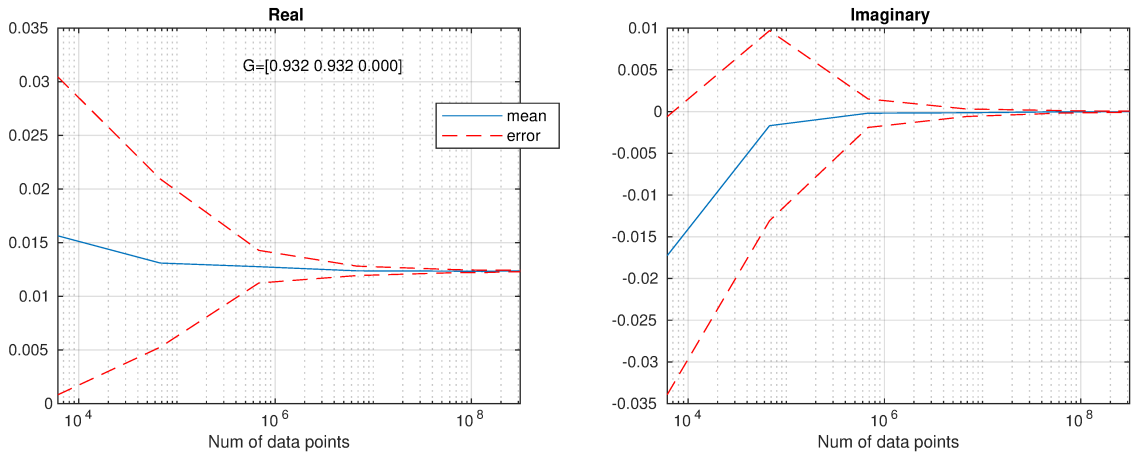

Figure 9: Real and imaginary part of the APMD as a function of data points for a carbon lattice when $\boldsymbol{G} = [0.932\ 0.932\ 0.000]$.

The left graph represents the real part of the complex number and the right graph the imaginary part. We can see that the imaginary part tends to zero just as predicted based on (125). Also the real part is clearly converging to a fixed value as well as the error margin is shrinking the more data points we have. We can deem this a reliable Monte Carlo simulation result.

However, this was only one example. In principle we need to examine these graphs for all the $\boldsymbol{G}$ points that we have. In the attachment B there are more graphs similar to figure 9 with different momentum values and for all 3 materials used in our simulations: carbon, silicon and germanium. Each row represents the APMD for different values of $\boldsymbol{G}$. Also for every $\pm\boldsymbol{G}$ pair, we can notice that the graphs are otherwise identical with the exception of being complex conjugates of each other. This makes sense considering our original formula for $\rho(\boldsymbol{G})$ defined in (117). From the graphs we can see that all the values are adequately converged at around $3 \cdot 10^7$ data points. Dividing that by $N = 64$, we get a suitable number for the amount of configurations for the CASINO input file, which is around 460000. Compared to the number of configurations we used (see 6.3), which is 960000, we can state that this is more than adequate. The graphs in B only shows examples for a handful $\boldsymbol{G}$-values. Because the $\boldsymbol{G}$-grid has thousands of points, we cannot possibly visually observe all of the convergences in a reasonable timeframe. By observing just a few quasi-randomly selected $\boldsymbol{G}$-points and observing their convergences, we can safely assume that the positron momentum density for all $\boldsymbol{G}$-points do in fact converge.

Considering the magnitudes of the real and imaginary part we can clearly see that the imaginary part always tends to zero no matter which $\boldsymbol{G}$ value is in question. This is in line with the inversion symmetry *i.e.*, $\rho(\boldsymbol{G}) = \rho(-\boldsymbol{G})$. From this point forth we can ignore the imaginary part completely and treat $\rho(\boldsymbol{G})$ as a real number.

7.2 Simulated results

Next we are going to visualize our simulated results in figure 10 along the high symmetry directions and compare them to the reference data from figure 8. In principle the data points for the high symmetry direction $\boldsymbol{d} = [d_1\ d_2\ d_3]$ are multiples of $\boldsymbol{d}$ for some coefficient t, *i.e.*,

$$\boldsymbol{G} + \boldsymbol{k}_s = t\boldsymbol{d} \quad \text{for some } t \in \mathbb{R}, \tag{161}$$

where $\boldsymbol{G} \in \mathbb{G}$. Since some $\boldsymbol{G}$ values have a randomly generated offset $\boldsymbol{k}_s$, it is better to choose a momentum value such that its distance from the line going through the origin and $\boldsymbol{d}$ is within a tolerance value r_{tol} *i.e.*,

$$\min_{t\in\mathbb{R}} |\boldsymbol{G} + \boldsymbol{k}_s - t\boldsymbol{d}| < r_{tol}. \tag{162}$$

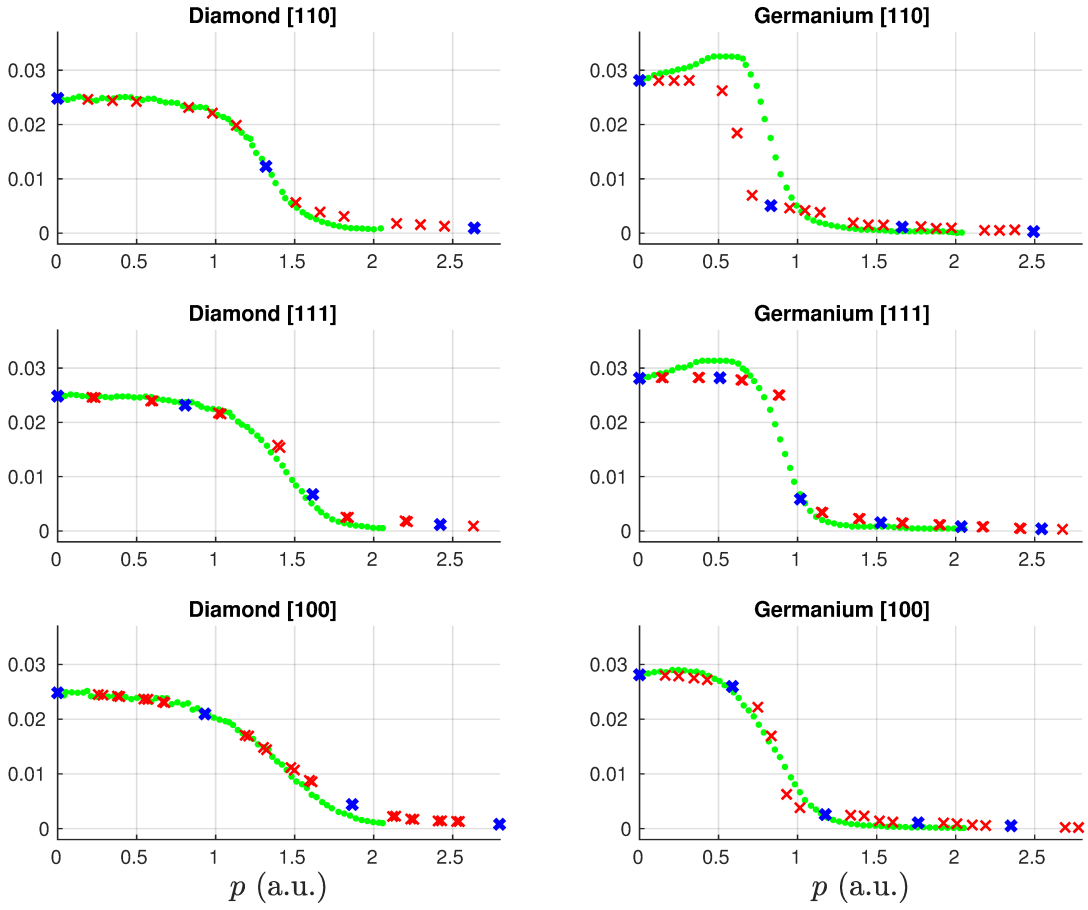

Figure 10: Green: reference data. Blue: simulated non-twisted data points. Red: simulated twisted data points.The data was fitted such that the values in the y-axis are the same at the origin. For the x-axis we multiplied the angular deviation in figure 8 with the 511 keV γ-ray's momentum in atomic units.

When comparing the simulation and reference data, for diamond, our simulations are very consistent with the reference data but not exactly the same, especially with larger momentum values. This may simply because of the approximation of the trial wave function and the fact that we haven't used enough parameters for the Jastrow factor. In the case of germanium, the computational and experimental data are not consistent. The reasons for this could be because of the pseudopotential for germanium. The pseudopotential used only approximates the electron potential for the 2 electrons in the $4s$ orbital in addition to the 2 electrons in the $4p$ orbital, while all the other electrons are treated as being "frozen" in place and their annihilation is completely neglected. One could argue that the $3d$ orbitals should be taking into account in the pseudopotential as well, since there are 10 of them and they too have a high contribution to the annihilation spectrum especially at high momenta like the $4s$ and $4p$ orbitals. That's why the results may deviate quite a lot compared to the experimental data. At least the data is consistent in the sense that the germanium data is more narrow than for silicon or diamond. Notice how the non-twisted data

points (blue crosses) alone would make it difficult to create a slope for the APMD. This is exactly why we used the twisted boundary conditions to the Bloch theorem in (50), *i.e.*, in order to get a better resolution for the data points (red crosses).

We have also simulated the results with silicon in figure 11, but the reference data [38] is made with DFT-based calculations [39]:

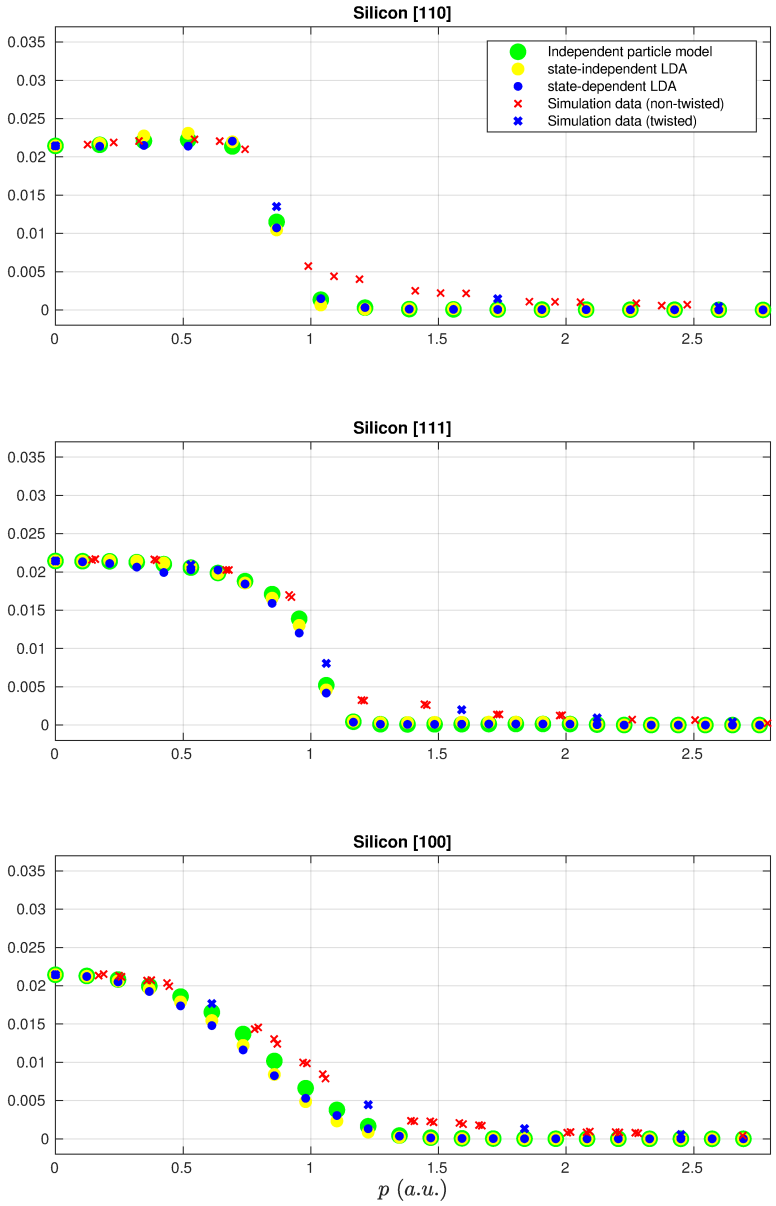

Figure 11: Comparing simulated data (blue and red crosses) with DFT-based calculations (green, yellow and blue dots) for silicon. The values have been normalized in a way that the values in the origin are the same.

The green dots come from the independent-particle model (IPM) which means that

no interactions between particles has been taken into account. The formula for the IPM is

$$\rho(\boldsymbol{p}) = \pi r_e^2 c \sum_j \left| \int e^{-i\boldsymbol{p}\cdot\boldsymbol{r}} \psi_+(\boldsymbol{r}) \psi_j(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r} \right|^2, \tag{163}$$

where r_e is the classical electron radius, ψ_+ is the wave function for the positron and ψ_j is the wave function for the electron on orbital j. The VMC sampling for the APMD without the Jastrow factor is formally the same as in (163) and we have empirically discovered that they're in fact the same within the error margin. The yellow dots represent the state-independent LDA (local density approximation) [40]. It attempts to take the electron-positron correlation effects into account with an enhancement factor $\sqrt{\gamma(n_-(\boldsymbol{r}))}$. The formula for the state-independent LDA is

$$\rho(\boldsymbol{p}) = \pi r_e^2 c \sum_j \left| \int e^{-i\boldsymbol{p}\cdot\boldsymbol{r}} \psi_+(\boldsymbol{r}) \psi_j(\boldsymbol{r}) \sqrt{\gamma(n_-(\boldsymbol{r}))} \, \mathrm{d}\boldsymbol{r} \right|^2. \tag{164}$$

Finally the blue dots represent the state-dependent LDA [41]. It uses a constant electron state dependent enhancement factor γ_j. The formula is

$$\rho(\boldsymbol{p}) = \pi r_e^2 c \sum_j \gamma_j \left| \int e^{-i\boldsymbol{p}\cdot\boldsymbol{r}} \psi_+(\boldsymbol{r}) \psi_j(\boldsymbol{r}) \, \mathrm{d}\boldsymbol{r} \right|^2, \tag{165}$$

such that γ_j is the ratio of the annihilation rate for the state j and for IPM *i.e.*, $\gamma_j = \lambda_j / \lambda_{ipm}$. Our results properly take into account the correlation effects between particles with the Jastrow factor, instead of using some enhancement factors that merely attempt to take correlation effects into account like in the DFT-based models. We can also see that our results differ quite a bit from the DFT-based results. We can conclude that the Jastrow factor is the most significant element explaining the differences of our simulations and the DFT-based reference data.

We have also simulated the data in 2D for C, Si and Ge in figures 12, 13 and 14 respectively, such that the results are taken from the plane formed by orthogonal basis vectors [001] and [110].

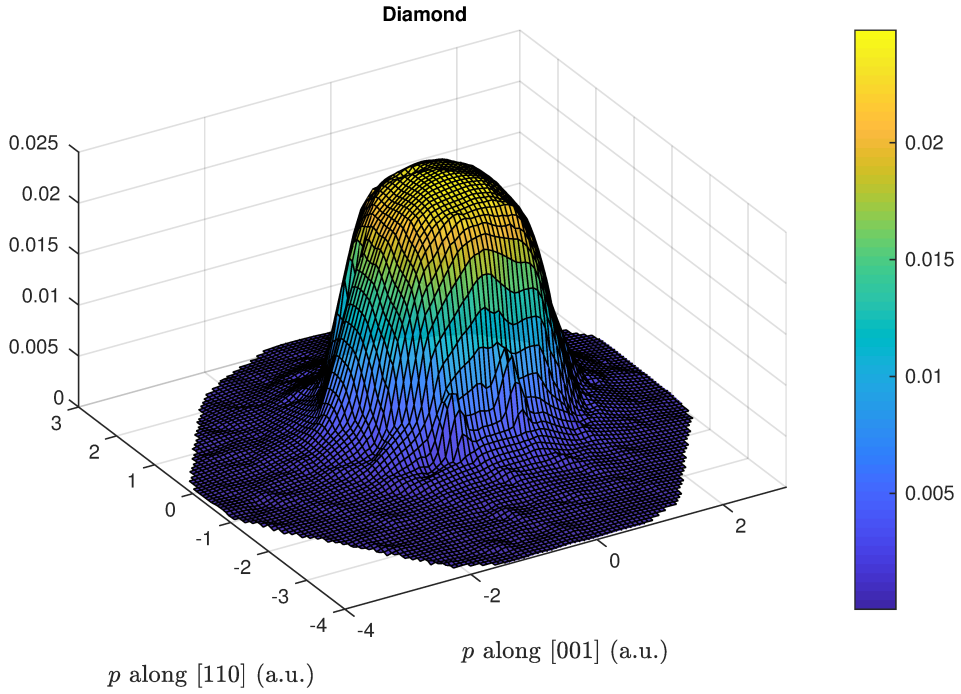

Figure 12: APMD in plane with basis vectors [001] and [110] for diamond

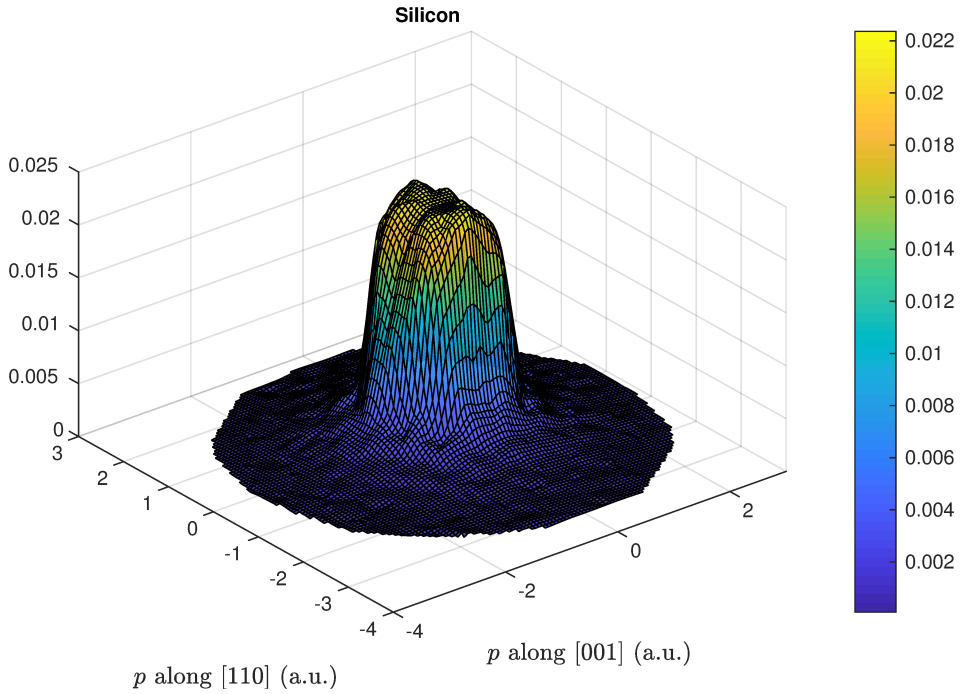

Figure 13: APMD in plane with basis vectors [001] and [110] for silicon

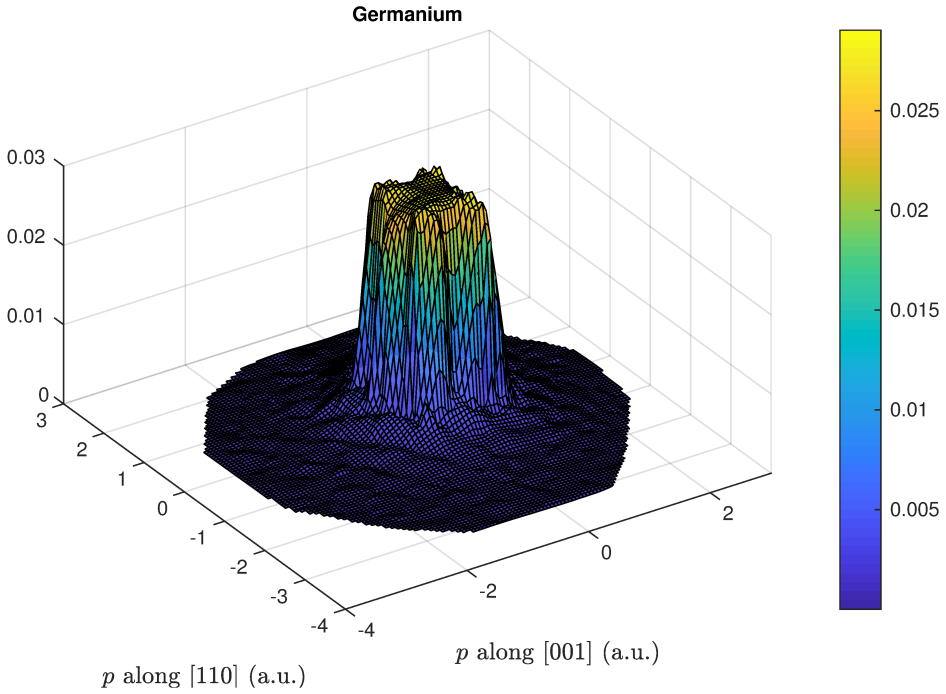

Figure 14: APMD in plane with basis vectors [001] and [110] for germanium

Here we can also see that the more heavy the material is, the more narrow the APMD gets. The surfaces for diamond and silicon look smooth enough, but not so much for germanium. The reasons for this are probably the same as in the 1D graphs; the pseudopotential is not suitable for APMD calculations.

8 Conclusions

We have developed a method to calculate the positron-electron annihilation momentum density in a periodic crystal for basically any momentum value, not just with discrete values in the reciprocal space. Just by changing the twisted boundary conditions, the twist averaging method lets us have a denser set of data points for the positron momentum density $\rho(\boldsymbol{G})$ that clearly makes a consistent slope. The data set would have been far more difficult to examine if we didn't use twist averaging. The non-twisted points are the blue datapoints in figures 10 and 11, which are clearly too few and far between.

The APMD is directly linked to the measurements of the Doppler broadening of the annihilation radiation, as well as the angular deviation distribution between the γ-rays upon annihilation. Our simulations are useful to be compared with experimental measurements in order to evaluate the accuracy of our wave function approximation. It is important to create a reliable way of simulating momentum densities, since for experimental measurements we don't know exactly what to expect and it is difficult to acquire non-defected reference samples for new materials, whereas

in the simulations we have an accurate model to work with. Also we can make simulations without the errors involved in DFT-based approximations.

In this project we only simulated with perfect lattices without any defects. One thing that could be relevant in future research would be to simulate the APMD with imperfect lattice structures with different kinds of point defects. Doing so and comparing them with experimental measurements with known defect structures would be of interest, but out of the scope of this project.

References

[1] R. Krause-Rehberg and H.S. Leipner. *Positron Annihilation in Semiconductors*. Springer-Verlag, 1999.

[2] M. Lannoo and J. Bourgoin. *Point Defects in Semiconductors I: Theoretical Aspects*. Springer Science Business Media, 1981.

[3] K. Saarinen, P. Hautojärvi, P. Lanki, and C. Corbel. Ionization levels of as vacancies in as-grown gaas studied by positron-lifetime spectroscopy. *Phys. Rev. B*, 44:10585–10600, Nov 1991.

[4] Filip Tuomisto and Ilja Makkonen. Defect identification in semiconductors with positron annihilation: Experiment and theory. *Rev. Mod. Phys.*, 85:1583–1631, Nov 2013.

[5] Dale L. Bailey, Joel S. Karp, and Suleman Surti. *Physics and Instrumentation in PET*, pages 13–39. Springer London, London, 2005.

[6] W. M. C. Foulkes, L. Mitas, R. J. Needs, and G. Rajagopal. Quantum monte carlo simulations of solids. *Rev. Mod. Phys.*, 73:33–83, Jan 2001.

[7] N. D. Drummond, P. López Ríos, R. J. Needs, and C. J. Pickard. Quantum monte carlo study of a positron in an electron gas. *Phys. Rev. Lett.*, 107:207402, Nov 2011.

[8] R J Needs, M D Towler, N D Drummond, and P López Ríos. Continuum variational and diffusion quantum monte carlo calculations. *Journal of Physics: Condensed Matter*, 22(2):023201, dec 2009.

[9] E. Boroński and R. M. Nieminen. Electron-positron density-functional theory. *Phys. Rev. B*, 34:3820–3831, Sep 1986.

[10] Richard M. Martin. *Electronic structure: Basic Theory and Practical Methods*. Cambridge University Press, Nov 2004.

[11] S. R. Elliott. *The Physics and Chemistry of Solids*. Wiley, 1998.

[12] J. R. Oppenheimer M. Born. *Zur quantentheorie der molekeln*, volume 74. Wiley-VCH, Aug 1927.

[13] Yukiumi Kita, Ryo Maezono, Masanori Tachikawa, Mike Towler, and Richard J. Needs. Ab initio quantum monte carlo study of the positronic hydrogen cyanide molecule. *The Journal of Chemical Physics*, 131(13):134310, 2009.

[14] G. Rajagopal, R. J. Needs, S. Kenny, W. M. C. Foulkes, and A. James. Quantum monte carlo calculations for solids using special k points methods. *Phys. Rev. Lett.*, 73:1959–1962, Oct 1994.

[15] G. Rajagopal, R. J. Needs, A. James, S. D. Kenny, and W. M. C. Foulkes. Variational and diffusion quantum monte carlo calculations at nonzero wave vectors: Theory and application to diamond-structure germanium. *Phys. Rev. B*, 51:10591–10600, Apr 1995.

[16] C. Lin, F. H. Zong, and D. M. Ceperley. Twist-averaged boundary conditions in continuum quantum monte carlo algorithms. *Phys. Rev. E*, 64:016702, Jun 2001.

[17] G. G. Ryzhikh and J. Mitroy. Positron annihilation profiles for HPs and He(3Se)e^+. *Journal of Physics B: Atomic, Molecular and Optical Physics*, 32(16):4051–4064, aug 1999.

[18] https://en.wikipedia.org/wiki/Pseudopotential.

[19] D. R. Hamann, M. Schlüter, and C. Chiang. Norm-conserving pseudopotentials. *Phys. Rev. Lett.*, 43:1494–1497, Nov 1979.

[20] Phillip A. Christiansen, Yoon S. Lee, and Kenneth S. Pitzer. Improved ab initio effective core potentials for molecular calculations. *The Journal of Chemical Physics*, 71(11):4445–4450, 1979.

[21] M Krauss and W J Stevens. Effective potentials in molecular quantum chemistry. *Annual Review of Physical Chemistry*, 35(1):357–385, 1984.

[22] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[23] Robert Jastrow. Many-body problem with strong forces. *Phys. Rev.*, 98:1479–1484, Jun 1955.

[24] N. D. Drummond, M. D. Towler, and R. J. Needs. Jastrow correlation factor for atoms, molecules, and solids. *Phys. Rev. B*, 70:235119, Dec 2004.

[25] Kristoffer Simula. Quantum monte carlo simulation of positrons in solids; kvantti monte carlo -simulointi positroneille kiinteässä aineessa. G2 pro gradu, diplomityö, 2017-10-03.

[26] P Giannozzi, O Andreussi, T Brumme, O Bunau, M Buongiorno Nardelli, M Calandra, R Car, C Cavazzoni, D Ceresoli, M Cococcioni, N Colonna, I Carnimeo, A Dal Corso, S de Gironcoli, P Delugas, R A DiStasio Jr, A Ferretti, A Floris, G Fratesi, G Fugallo, R Gebauer, U Gerstmann, F Giustino, T Gorni, J Jia, M Kawamura, H-Y Ko, A Kokalj, E Küçükbenli, M Lazzeri, M Marsili, N Marzari, F Mauri, N L Nguyen, H-V Nguyen, A Otero de-la Roza, L Paulatto, S Poncé, D Rocca, R Sabatini, B Santra, M Schlipf, A P Seitsonen, A Smogunov, I Timrov, T Thonhauser, P Umari, N Vast, X Wu, and S Baroni. Advanced capabilities for materials modelling with quantum espresso. *Journal of Physics: Condensed Matter*, 29(46):465901, 2017.

[27] Paolo Giannozzi, Stefano Baroni, Nicola Bonini, Matteo Calandra, Roberto Car, Carlo Cavazzoni, Davide Ceresoli, Guido L Chiarotti, Matteo Cococcioni, Ismaila Dabo, Andrea Dal Corso, Stefano de Gironcoli, Stefano Fabris, Guido Fratesi, Ralph Gebauer, Uwe Gerstmann, Christos Gougoussis, Anton Kokalj, Michele Lazzeri, Layla Martin-Samos, Nicola Marzari, Francesco Mauri, Riccardo Mazzarello, Stefano Paolini, Alfredo Pasquarello, Lorenzo Paulatto, Carlo Sbraccia, Sandro Scandolo, Gabriele Sclauzero, Ari P Seitsonen, Alexander Smogunov, Paolo Umari, and Renata M Wentzcovitch. Quantum espresso: a modular and open-source software project for quantum simulations of materials. *Journal of Physics: Condensed Matter*, 21(39):395502 (19pp), 2009.

[28] Paolo Giannozzi, Stefano Baroni, Nicola Bonini, Matteo Calandra, Roberto Car, Carlo Cavazzoni, Davide Ceresoli, Guido L Chiarotti, Matteo Cococcioni, Ismaila Dabo, Andrea Dal Corso, Stefano de Gironcoli, Stefano Fabris, Guido Fratesi, Ralph Gebauer, Uwe Gerstmann, Christos Gougoussis, Anton Kokalj, Michele Lazzeri, Layla Martin-Samos, Nicola Marzari, Francesco Mauri, Riccardo Mazzarello, Stefano Paolini, Alfredo Pasquarello, Lorenzo Paulatto, Carlo Sbraccia, Sandro Scandolo, Gabriele Sclauzero, Ari P Seitsonen, Alexander Smogunov, Paolo Umari, and Renata M Wentzcovitch. QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials. *Journal of Physics: Condensed Matter*, 21(39):395502, sep 2009.

[29] https://www.quantum espresso.org/.

[30] https://vallico.net/casinoqmc/.

[31] J. R. Trail and R. J. Needs. Smooth relativistic hartree–fock pseudopotentials for h to ba and lu to hg. *The Journal of Chemical Physics*, 122(17):174109, 2005.

[32] J. R. Trail and R. J. Needs. Norm-conserving hartree–fock pseudopotentials and their asymptotic behavior. *The Journal of Chemical Physics*, 122(1):014112, 2005.

[33] John P. Perdew, Kieron Burke, and Matthias Ernzerhof. Generalized gradient approximation made simple. *Phys. Rev. Lett.*, 77:3865–3868, Oct 1996.

[34] T. Torsti, T. Eirola, J. Enkovaara, T. Hakala, P. Havu, V. Havu, T. Höynälänmaa, J. Ignatius, M. Lyly, I. Makkonen, T. T. Rantala, J. Ruokolainen, K. Ruotsalainen, E. Räsänen, H. Saarikoski, and M. J. Puska. Three real-space discretization techniques in electronic structure calculations. *physica status solidi (b)*, 243(5):1016–1053, 2006.

[35] D. Alfè and M. J. Gillan. Efficient localized basis set for quantum Monte Carlo calculations on condensed matter. *Phys. Rev. B*, 70:161101, Oct 2004.

[36] W. Liu, S. Berko, and Allen P. Mills. Electron-positron momentum density in ge and diamond. In *Positron Annihilation - ICPA-9*, volume 105 of *Materials Science Forum*, pages 743–746. Trans Tech Publications Ltd, 1 1992.

[37] Yuri Nakao, Tadaharu Shibuya, Shin Takeuchi, Weimin Liu, Xue-Song Li, and Stephan Berko. Positron-annihilation studies of a decagonal single quasicrystal of $Al_{70}Ni_{15}Co_{15}$. *Phys. Rev. B*, 46:3108–3111, Aug 1992.

[38] I. Makkonen. unpublished.

[39] I. Makkonen, M. Hakala, and M. J. Puska. Modeling the momentum distributions of annihilating electron-positron pairs in solids. *Phys. Rev. B*, 73:035103, Jan 2006.

[40] S Daniuk, G Kontrym-Sznajd, A Rubaszek, H Stachowiak, J Mayers, P A Walters, and R N West. Selective enhancement of different electron populations by electron-positron attraction: application to zinc. *Journal of Physics F: Metal Physics*, 17(6):1365–1378, jun 1987.

[41] M. Alatalo, B. Barbiellini, M. Hakala, H. Kauppinen, T. Korhonen, M. J. Puska, K. Saarinen, P. Hautojärvi, and R. M. Nieminen. Theoretical and experimental study of positron annihilation with core electrons in solids. *Phys. Rev. B*, 54:2397–2409, Jul 1996.

A Input and output files

This attachment has all the relevant input and output files used in our simulation process which is compiled in figure 7. These files are used with PWscf, Atsup and CASINO. If the files are to long for this, we will use a shortened version with the label "sketch". Whenever there is the symbol `[...]` in a sketch, note that this is only there to shorten the file.

Filename: `awfn.data` (sketch)

```
Atomic C wave function in real space
Atomic number
 6
Total number of orbitals
 2
The 2s(2)2p(2) [3P] state electronic configuration
Number of up, downspin electrons
 3  1
States
 1  2  0  0
 2  2  1  1
 3  2  1  0
 1  2  0  0
Radial grid (a.u.)
 1539
    0.000000000000000E+00
    0.167782732554280E-08
    0.337817548736032E-08
    0.510134677404852E-08
    0.684764753170787E-08
    0.861738821840578E-08
[...]
Orbital # 1 [2s]
0 2 0
    0.000000000000000E+00
    0.889411457743842E-09
    0.179076114626732E-08
    0.270420930788899E-08
    0.362991833579894E-08
    0.456805280293644E-08
[...]
Orbital # 2 [2p]
0 2 1
    0.000000000000000E+00
    0.821600888099756E-17
    0.333066371897552E-16
    0.759514495415207E-16
    0.136851376299690E-15
    0.216729331534892E-15
[...]
```

Filename: `c_pp.data` (sketch)

```
DF Pseudopotential in real space for C
Atomic number and pseudo-charge
6 4.00
Energy units (rydberg/hartree/ev):
rydberg
Angular momentum of local component (0=s,1=p,2=d..)
2
NLRULE override (1) VMC/DMC (2) config gen (0 ==> input/default value)
0 0
Number of grid points
1539
R(i) in atomic units
    0.000000000000000E+00
    0.167782732554280E-08
    0.337817548736032E-08
    0.510134677404852E-08
    0.684764753170787E-08
    0.861738821840578E-08
[...]
r*potential (L=0) in Ry
    0.000000000000000E+00
    0.229648517429911E-08
    0.462379519322339E-08
    0.698234380690572E-08
    0.937255031908937E-08
    0.117948396616717E-07
[...]
r*potential (L=1) in Ry
    0.000000000000000E+00
   -0.407286738878102E-07
   -0.820040332314775E-07
   -0.123833415981372E-06
   -0.166224258582373E-06
   -0.209184097295925E-06
[...]
r*potential (L=2) in Ry
    0.000000000000000E+00
   -0.321526585833896E-07
   -0.647368900400600E-07
   -0.977584872080890E-07
   -0.131223320680610E-06
   -0.165137339849492E-06
[...]
```

Filename: `inputpp`

```
&inputpp
        pp_data='c_pp.data'
        upf_file='c_df.UPF'
        /
        1
        awfn.data
```

Filename: c_df.UPF (sketch)

```
<?xml version="1.0" encoding="UTF-8"?>
<UPF version="">
  <PP_INFO><![CDATA[
From a Trail & Needs tabulated PP for CASINO
Author: unknown
Generation date: unknown
Pseudopotential type: NC
Element: C
Functional: HF
    Suggested minimum cutoff for wavefunctions:   0. Ry
    Suggested minimum cutoff for charge density:   0. Ry
    The Pseudo was generated with a Scalar-Relativistic Calculation
    L component and cutoff radius for Local Potential:  2   0.0000
    Valence configuration:
    nl pn  l   occ       Rcut    Rcut US       E pseu
    2s  2  0  2.00      0.000      0.000     0.000000
    2p  2  1  2.00      0.000      0.000     0.000000
    Generation configuration: not available.
    Comment:  Info: automatically converted from CASINO Tabulated format
]]></PP_INFO>
  <!--
 END OF HUMAN READABLE SECTION
-->
  <PP_HEADER generated="From a Trail &amp;amp; Needs tabulated PP for CASINO" \
  author="unknown
" date="unknown" comment="Info: automatically converted from CASINO Tabulated \
format" element="C " pseudo_type="NC" relativistic="scalar" \
is_ultrasoft="false" is_paw="false" is_coulomb="false" has_so="false" \
has_wfc="false" has_gipaw="false" paw_as_gipaw="false" \
core_correction="false" functional="HF" z_valence="4.000000000000e0" \
total_psenergy="0.000000000000e0" wfc_cutoff="0.000000000000e0" \
rho_cutoff="0.000000000000e0" l_max="1" l_max_rho="0" l_local="2" \
mesh_size="1539" number_of_wfc="2" number_of_proj="2"/>
  <PP_MESH dx="1.333333333333e-2" mesh="1539" xmin="-7.000000000000e0" \
  rmax="1.006565841037e2" zmesh="6.000000000000e0">
    <PP_R>
      0.000000000000000e0 1.677827325542800e-9 3.378175487360319e-9 \
      5.101346774048520e-9
      6.847647531707869e-9 8.617388218405777e-9 1.041088345936980e-8 \
      1.222845210292200e-8
      1.407041727716380e-8 1.593710644742190e-8 1.782885147446530e-8 \
      1.974598867350330e-8
[...]
```

Filename: `in.pwscf` (the flag `ecutwfc` has to be iterated)

```
&CONTROL
   prefix='carbon'
   restart_mode='from_scratch'
   calculation='scf'
   pseudo_dir='./'
   outdir='./'
   verbosity='high'
   wf_collect=.true.
 /
 &system
   input_dft="PBE"
   ibrav=0
   celldm(1) = 6.740652601
   nat= 2, ntyp= 1,
   ecutwfc = 220,
   nosym=.true.
   noinv=.true.
 /
 &electrons
   diagonalization='cg'
   conv_thr = 1.0d-10
   mixing_mode='plain'
   mixing_beta=0.7
   electron_maxstep=1000
 /
CELL_PARAMETERS alat
0.5 0.5 0.0
0.0 0.5 0.5
0.5 0.0 0.5
ATOMIC_SPECIES
C 12.0107 c_df.UPF
ATOMIC_POSITIONS crystal
C 0.00 0.00 0.00
C 0.25 0.25 0.25
K_POINTS crystal
8
 0     0      0    1
 0     0     -0.5  1
 0    -0.5    0    1
 0    -0.5   -0.5  1
-0.5   0      0    1
-0.5   0     -0.5  1
-0.5  -0.5    0    1
-0.5  -0.5   -0.5  1
```

Filename: `in.pp`

```
&inputpp
   prefix  = 'carbon'
   outdir = './'
   filplot = 'ccharge'
   plot_num= 0

/
&plot
   nx=200, ny=200, nz=200
   iflag =3
/
```

Filename: `ccharge` (sketch)

```
     48       48       48       48       48       48        2        1
    0         6.74065260       0.00000000       0.00000000       0.00000000 \
    0.00000000       0.00000000
  0.50000000000000000        0.50000000000000000         0.0000000000000000
   0.0000000000000000        0.50000000000000000        0.50000000000000000
  0.50000000000000000         0.0000000000000000        0.50000000000000000
    1012.8073062501         4.0000000000       220.0000000000     0
   1   C     4.00
   1       0.000000000    0.000000000    0.000000000    1
   2       0.250000000    0.250000000    0.250000000    1
  2.163199033E-02  4.288868473E-02  9.870210897E-02  1.684471987E-01 \
  2.285568957E-01
  2.645488977E-01  2.760372586E-01  2.711565637E-01  2.579592991E-01 \
  2.408430672E-01
  2.217656415E-01  2.020470718E-01  1.828654125E-01  1.650421027E-01 \
  1.489596776E-01
  1.347295508E-01  1.223491562E-01  1.117441465E-01  1.027890884E-01 \
  9.536084015E-02
  8.937517806E-02  8.477589924E-02  8.151639882E-02  7.956555766E-02 \
  7.891494100E-02
  7.956556278E-02  8.151640873E-02  8.477591333E-02  8.937519554E-02 \
  9.536086014E-02
[...]
```

Filename: `carbon.pwfn.data.n` (sketch) (if $n = 1$ the first k-point is just the γ-point)

```
BASIC INFO
----------
Generated by:
 PWSCF
Method:
 DFT
DFT Functional:
 unknown
Pseudopotential
 unknown
Plane wave cutoff (au)
  110.000000000000
Spin polarized:
F
Total energy (au per primitive cell)
 -11.3741172751764
Kinetic energy (au per primitive cell)
  8.48568096306558
Local potential energy (au per primitive cell)
 -5.13900259191072
Non local potential energy(au per primitive cell)
 0.660954029054864
Electron electron energy (au per primitive cell)
 0.957663035522627
Ion-ion energy (au per primitive cell)
 -12.7864130980751
Number of electrons per primitive cell
          8

GEOMETRY
--------
Number of atoms per primitive cell
          2
Atomic number and position of the atoms(au)
    6    0.00000000000000    0.00000000000000    0.00000000000000
    6    1.68516315025000    1.68516315025000    1.68516315025000
Primitive lattice vectors (au)
   3.370326300500000    3.370326300500000    0.000000000000000
   0.000000000000000    3.370326300500000    3.370326300500000
   3.370326300500000    0.000000000000000    3.370326300500000

G VECTORS
---------
```

```
 Number of G-vectors
        4735
 Gx Gy Gz (au)
    0.000000000000000    0.000000000000000    0.000000000000000
    0.932133085489000    0.932133085489000   -0.932133085489000
   -0.932133085489000   -0.932133085489000    0.932133085489000
[...]
 WAVE FUNCTION
 -------------
 Number of k-points
           8
 k-point # ; # of bands (up spin/down spin);            k-point coords (au)
   1    4    0  0.2956736602818729  0.0095528292901704  1.0729402814110671
 Band, spin, eigenvalue (au)
           1           1 -3.493240859230842E-002
 Eigenvectors coefficients
 (1.099819872344032E-005,2.803214049247994E-003)
 (6.636500164675435E-002,6.579392357300295E-002)
 (-3.650128465109637E-002,3.681827621235225E-002)
[...]
 k-point # ; # of bands (up spin/down spin);            k-point coords (au)
   8    4    0  0.7617402030263730  0.4756193720346706 -0.3252593468224334
 Band, spin, eigenvalue (au)
           1           1 -7.687683964783550E-002
 Eigenvectors coefficients
 (-0.257028985185180,0.199009933809533)
 (-3.200249255069077E-003,2.515404197112077E-002)
 (-0.898379242192726,-0.114298742961668)
[...]
```

Filename: `pwfn.data.n` (sketch) (if $n = 1$ the number of k-points is only 8)

```
C

BASIC INFO
----------
Generated by:
PWSCF                          MIKA doppler
Method:
DFT
DFT Functional
unknown
Pseudopotential
unknown
Plane wave cutoff (au)
  110.00000000000000       (electrons)
Spin polarized:
F
Total energy (au per primitive cell)
 -11.374117275176401
Kinetic energy (au per primitive cell)
  8.4856809630693792
Local potential energy (au per primitive cell)
 -5.1390025919106996
Non-local potential energy (au per primitive cell)
 0.66095402905216605
Electron-electron energy (au per primitive cell)
 0.95766303552088905
Ion-ion energy (au per primitive cell)
 -12.786413098075100
Number of electrons per primitive cell
          8

GEOMETRY
--------
Number of atoms per primitive cell
          2
Atomic Numbers and positions of atoms (au)
          6   0.0000000000000000        0.0000000000000000        \
0.0000000000000000
          6   1.6851631502500000        1.6851631502500000        \
1.6851631502500000
Primitive lattice vectors (au)
  3.3703263004999999        3.3703263004999999        0.0000000000000000
  0.0000000000000000        3.3703263004999999        3.3703263004999999
  3.3703263004999999        0.0000000000000000        3.3703263004999999
```

```
 G VECTORS
 ---------
 Number of G-vectors
       11935
 Gx Gy Gz (au)
     0.000000000000000     0.000000000000000     0.000000000000000
     0.932133085627286     0.932133085627286    -0.932133085627286
    -0.932133085627286    -0.932133085627286     0.932133085627286
[...]
 WAVE FUNCTION
 -------------
 Number of k-points
           9
 k-point # ; # of bands (up spin/down spin) ; k-point coords (au)
           1           1           0 0.00000 0.00000 0.00000
 Band, spin, eigenvalue (au)
           1           1   9999999.00
 Eigenvector coeficients
 (  0.0000000000000000     ,  0.0000000000000000     )
 (  0.0000000000000000     ,  0.0000000000000000     )
 (  0.0000000000000000     ,  0.0000000000000000     )
[...]
 k-point # ; # of bands (up spin/down spin) ; k-point coords (au)
           2           4           0  0.29567366028187292   \
9.5528292901704004E-003   1.0729402814110671
 Band, spin, eigenvalue (au)
           1           1  -3.4932408592476739E-002
 Eigenvector coeficients
   (1.0998198611881849E-005,2.8032140492674699E-003)
   (6.6365001646636834E-002,6.5793923572521603E-002)
  (-3.6501284651448251E-002,3.6818276212583982E-002)
[...]
 k-point # ; # of bands (up spin/down spin) ; k-point coords (au)
           9           4           0  0.76174020302637302       \
0.47561937203467058      -0.32525934682243340
 Band, spin, eigenvalue (au)
           1           1  -7.6876839648006012E-002
 Eigenvector coeficients
            (-0.25702898518510697,0.19900993381026000)
  (-3.2002492549939132E-003,2.5154041970799741E-002)
           (-0.89837924219312504,-0.11429874296190599)

[...]
 k-point # ; # of bands of positron ; k-point coords (au)
```

```
  1           1   0.0000000000000000        0.0000000000000000  \
0.0000000000
 Band, spin, eigenvalue (au)
          1           3           0
 Eigenvector coefficients
             (0.96955920861530021,0.0000000000000000)
  (-5.90195324402546170E-002,5.90408919573138166E-002)
 (-5.90195324402546170E-002,-5.90408919573138166E-002)

[...]
```

Filename: `bwfn.data.n` (sketch) (if $n = 1$ the number of k-points is only 8)

```
C

BASIC INFO
----------
Generated by:
 PWSCF                          MIKA doppler
Method:
 DFT
DFT Functional:
 unknown
Pseudopotential
 unknown
Plane wave cutoff (au)
  110.000000000000
Spin polarized:
F
Total energy (au per primitive cell)
 -11.3741172751764
Kinetic energy (au per primitive cell)
  8.48568096306938
Local potential energy (au per primitive cell)
 -5.13900259191070
Non local potential energy(au per primitive cell)
 0.660954029052166
Electron electron energy (au per primitive cell)
 0.957663035520889
Ion-ion energy (au per primitive cell)
 -12.7864130980751
Number of electrons per primitive cell
          8

GEOMETRY
--------
Number of atoms per primitive cell
          2
Atomic number and position of the atoms(au)
6  0.00000000000000000E+00  0.00000000000000000E+00  0.00000000000000000E+00
6  1.68516315024999996E+00  1.68516315024999996E+00  1.68516315024999996E+00
Primitive lattice vectors (au)
 3.37032630049999993E+00  3.37032630049999993E+00  0.00000000000000000E+00
 0.00000000000000000E+00  3.37032630049999993E+00  3.37032630049999993E+00
 3.37032630049999993E+00  0.00000000000000000E+00  3.37032630049999993E+00

G VECTORS
```

```
 ---------
 Number of G-vectors
       11935
 Gx Gy Gz (au)
  0.0000000000000000E+00  0.0000000000000000E+00  0.0000000000000000E+00
  9.3213308562728602E-01  9.3213308562728602E-01 -9.3213308562728602E-01
 -9.3213308562728602E-01 -9.3213308562728602E-01  9.3213308562728602E-01
[...]
 -1.0253463941900144E+01  8.3891977706455734E+00 -1.0253463941900144E+01
 Blip grid
 92 92 92

 WAVE FUNCTION
 -------------
 Number of k-points
          9
 k-point # ; # of bands (up spin/down spin) ; k-point coords (au)
 1 1 0  0.0000000000000000E+00  0.0000000000000000E+00 \
 0.0000000000000000E+00
 Band, spin, eigenvalue (au), localized
 1 1  9.9999990000000000E+06 F
 Complex blip coefficients for extended orbital
 (0.000000000000000E+000,0.000000000000000E+000)
 (0.000000000000000E+000,0.000000000000000E+000)
 (0.000000000000000E+000,0.000000000000000E+000)
[...]
 k-point # ; # of bands (up spin/down spin) ; k-point coords (au)
 2 4 0  2.9567366028187292E-01  9.5528292901704004E-03 \
 1.0729402814110671E+00
 Band, spin, eigenvalue (au), localized
 1 1 -3.4932408592476739E-02 F
 Complex blip coefficients for extended orbital
 (0.213885965003784,8.962525303196627E-004)
 (0.213432813524621,-4.048884120418360E-002)
 (0.211991868556838,-8.120847410560997E-002)
[...]
 k-point # ; # of bands (up spin/down spin) ; k-point coords (au)
 9 4 0  7.6174020302637302E-01  4.7561937203467058E-01 \
 -3.2525934682243340E-01
 Band, spin, eigenvalue (au), localized
 1 1 -7.6876839648006012E-02 F
 Complex blip coefficients for extended orbital
 (-0.219190693448725,-1.904452458676376E-003)
 (-0.225191212622335,-2.414929647739128E-002)
 (-0.234471423212596,-4.530395997712052E-002)
```

[...]

Filename: `correlation.data` (sketch)

```
START HEADER
No title given.
END HEADER

START VERSION
  1
END VERSION

START JASTROW
Title
Carbon 2x2x2 (PW basis)
Truncation order C
  3
START U TERM
Number of sets
  1
START SET 1
Spherical harmonic l,m
  0 0
Expansion order N_u
  16
Spin dep (0->uu=dd=ud; 1->uu=dd/=ud; 2->uu/=dd/=ud)
  2
Cutoff (a.u.)     ;  Optimizable (0=NO; 1=YES)
  4.76635639484821                0
Parameter values  ;  Optimizable (0=NO; 1=YES)
 3.302249481056885E-003           1       ! alpha_0,1
[...]
-6.814717574888992E-010           1       ! alpha_16,4
END SET 1
END U TERM
START CHI TERM
Number of sets ; labelling (1->atom in s. cell; 2->atom in p. cell; 3->species)
  1 3
START SET 1
Spherical harmonic l,m
  0 0
Number of species in set
  1
Label of the species in this set
   1
Impose electron-nucleus cusp (0=NO; 1=YES)
  0
Expansion order N_chi
```

```
   12
 Spin dep (0->u=d; 1->u/=d)
   1
 Cutoff (a.u.)      ;  Optimizable (0=NO; 1=YES)
   3.67513288983045                0
 Parameter values   ;  Optimizable (0=NO; 1=YES)
 -7.130888391451106E-003           1       ! beta_0,1,1
[...]
  9.009535231577942E-006           1       ! beta_12,2,1
 END SET 1
 END CHI TERM
 START F TERM
 Number of sets ; labelling (1->atom in s. cell; 2->atom in p. cell; 3->species)
   1 3
 START SET 1
 Number of species in set
   1
 Label of the species in this set
    1
 Prevent duplication of u term (0=NO; 1=YES)
   0
 Prevent duplication of chi term (0=NO; 1=YES)
   0
 Electron-nucleus expansion order N_f_eN
   2
 Electron-electron expansion order N_f_ee
   2
 Spin dep (0->uu=dd=ud; 1->uu=dd/=ud; 2->uu/=dd/=ud)
   2
 Cutoff (a.u.)      ;  Optimizable (0=NO; 1=YES)
   2.37847922575693                0
 Parameter values   ;  Optimizable (0=NO; 1=YES)
 -5.290758673338731E-004           1       ! gamma_1,1,0,1,1
[...]
  3.289778648279989E-004           1       ! gamma_2,2,2,4,1
 END SET 1
 END F TERM
 END JASTROW
```

Filename: input

```
#------------------#
# CASINO input file #
#------------------#

# 2x2x2 (plane wave basis)

# SYSTEM
neu                : 32            #*! Number of up electrons (Integer)
ned                : 32            #*! Number of down electrons (Integer)
nhu                : 1
periodic           : T             #*! Periodic boundary conditions (Boolean)
atom_basis_type    : blip          #*! Basis set type (Text)
%block npcell
2 2 2
%endblock npcell
%block particles
3 1 1 0.5 Spin-up positron
%endblock particles

psi_s              : slater        #*! Type of [anti]symmetrizing wfn (Text)
complex_wf         : T             #*! Wave function real or complex (Boolean)

# RUN
runtype            : vmc           #*! Type of calculation (Text)
newrun             : T             #*! New run or continue old (Boolean)
testrun            : F             #*! Test run flag (Boolean)
block_time         : 0.0 s         #*! VMC/DMC block time (Physical)


# VMC
vmc_equil_nstep    : 20000        #*! Number of equilibration steps (Integer)
vmc_nstep          : 960000          #*! Number of steps (Integer)
vmc_nblock         : 1             #*! Number of checkpoints (Integer)
vmc_nconfig_write  : 1            #*! Number of configs to write (Integer)
vmc_decorr_period  : 20
writeout_vmc_hist  : T

# DMC
dmc_equil_nstep    : 2000          #*! Number of steps (Integer)
dmc_equil_nblock   : 1             #*! Number of checkpoints (Integer)
dmc_stats_nstep    : 256           #*! Number of steps (Integer)
dmc_stats_nblock   : 1             #*! Number of checkpoints (Integer)
dmc_target_weight  : 1000.d0       #*! Total target weight in DMC (Real)
dtdmc              : 0.003         #*! DMC time step (Real)
```

```
use_tmove         : F              #*! Casula nl pp for DMC (Boolean)

# RMC

# OPTIMIZATION
opt_method        : varmin         #*! Opt method (varmin/madmin/emin/...)
opt_cycles        : 4              #*! Number of optimization cycles (Integer)
opt_jastrow       : T              #*! Optimize Jastrow factor (Boolean)
opt_det_coeff     : F              #*! Optimize determinant coeffs (Boolean)
opt_backflow      : F              #*! Optimize backflow parameters (Boolean)
opt_orbitals      : F              #*! Optimize orbital parameters (Boolean)
opt_info          : 5              #*! Amount of information from minimization

# GENERAL PARAMETERS
use_jastrow       : T              #*! Use a Jastrow function (Boolean)
backflow          : F              #*! Use backflow corrections (Boolean)
expot             : F              #*! Use external potential (Boolean)
timing_info       : F              #*! Activate subroutine timers (Boolean)
esupercell        : F              #*! Energy/supercell in output (Boolean)
neighprint        : 0              #*! Neighbour analysis (Integer)
mpc_cutoff        : 30.d0 hartree  #*! G vector cutoff for MPC (Physical)
interaction       : ewald          #*! Interaction type (Text)
finite_size_corr  : F              #*! Eval. finite size correction (Boolean)
forces            : F              #*! Evaluate forces on atoms (Boolean)
checkpoint        : 1              #*! Checkpoint level (Integer)
hartree_xc        : F              #*! XC and Hartree if SF or MPC (Boolean)
pn_cusp           : 0
positron_pp       : T

# EXPECTATION VALUES
pos_mom_den       : T
pair_density      : F              #*! Accumulate pairt density for \
enhancement factor (Boolean)
density           : F              #*! Accumulate density (Boolean)
spin_density      : F              #*! Accumulate spin densities (Boolean)
pair_corr         : F              #*! Accumulate rec. space PCF (Boolean)
pair_corr_sph     : F              #*! Accumulate sph. real space PCF (Boolean)
loc_tensor        : F              #*! Accumulate localization tensor (Boolean)
structure_factor  : F              #*! Accumulate structure factor (Boolean)
struc_factor_sph  : F              #*! Accumulate sph. struc. factor (Boolean)
onep_density_mat  : F              #*! Accumulate 1p density matrix (Boolean)
twop_density_mat  : F              #*! Accumulate 2p density matrix (Boolean)
cond_fraction     : F              #*! Accumulate cond fraction (Boolean)
twop_dm_mom       : F              #*! Accum 2p momentum density (Boolean)
cond_fraction_mom : F              #*! Accum strict 2p momentum density (Boo...
```

```
dipole_moment     : F              #*! Accumulate elec. dipole moment (Boolean)
expval_cutoff     : 4.d0 hartree  #*! G vector cutoff for expval (Physical)
permit_den_symm   : F              #*! Symmetrize QMC charge data (Boolean)
qmc_density_mpc   : F              #*! Use QMC density in MPC int (Boolean)
contact_den       : F
```

Filename: `expval.data` (sketch)

```
START HEADER
 No header text
END HEADER

START EXPVAL
Title
 No title given
File version
 1
Number of particle types (e.g. 2=electrons, 4=electrons+holes)
 3
Number of each type of particle
 32 32 1
Dimensionality
 3
Periodicity
 3
Primitive translation vectors (au)
  3.3703263004999999E+00  3.3703263004999999E+00  0.0000000000000000E+00
  0.0000000000000000E+00  3.3703263004999999E+00  3.3703263004999999E+00
  3.3703263004999999E+00  0.0000000000000000E+00  3.3703263004999999E+00
Supercell matrix (11),(22),(33),(12),(13),(21),(23),(31),(32)
 2 2 2 0 0 0 0 0 0
Volume of simulation cell
 6.1254194180659476E+02
Radius of sphere inscribed in Wigner-Seitz cell of simulation cell
 4.7663611637898393E+00
Number of available G-vector sets
 1

START GVECTOR SET 1
Energy cutoff (au) used to generate set
 6.0000000000000000E+00
Number of G-vectors in set
 411
Supercell reciprocal lattice vectors (au)
  4.6606654274450021E-01  4.6606654274450021E-01 -4.6606654274450021E-01
 -4.6606654274450021E-01  4.6606654274450021E-01  4.6606654274450021E-01
  4.6606654274450021E-01 -4.6606654274450021E-01  4.6606654274450021E-01
G-vector components Gx, Gy, Gz (au)
  0.0000000000000000E+00  0.0000000000000000E+00  0.0000000000000000E+00
 -4.6606654274450021E-01  4.6606654274450021E-01 -4.6606654274450021E-01
  4.6606654274450021E-01 -4.6606654274450021E-01  4.6606654274450021E-01
[...]
```

```
 -1.1102230246251565E-16  2.7963992564670015E+00  1.8642661709780008E+00
END GVECTOR SET 1

START POSITRON MOMENTUM DENSITY
Accumulation carried out using
 VMC
Use G-vector set
 1
Number of sets
 1
START SET 1
Total weight
  0.000000000000000E+000
Complex pair-density coefficients (real part, imaginary part)
   1210516.40607896       -3508.60371169751
   211333.947750287        42.8190681535057
   1429598.10981610        5606.42916629981
[...]
   46300.1350183514       -463.172085603084
END SET 1
END POSITRON MOMENTUM DENSITY

START POSITRON MOMENTUM DENSITY SQUARED
Accumulation carried out using
 VMC
Use G-vector set
 1
Number of sets
 1
START SET 1
Total weight
  0.000000000000000E+000
Complex pair-density coefficients (real part, imaginary part)
   20236804.7842913        20082357.7877776
   20177867.6346800        20141294.9373888
   20279621.1845771        20039541.3874918
[...]
   20285782.4052097        20033380.1668593
END SET 1
END POSITRON MOMENTUM DENSITY SQUARED
END EXPVAL
```

B Convergence

In this attachment there are a few graphs visualising how the APMD $\rho(\boldsymbol{G})$ converges with different parameter values $\boldsymbol{G}$ as a function of the configurations used for the simulations. The error margin is the 95% confidence interval, which is equal to twice the sample standard deviation of the mean in each direction.

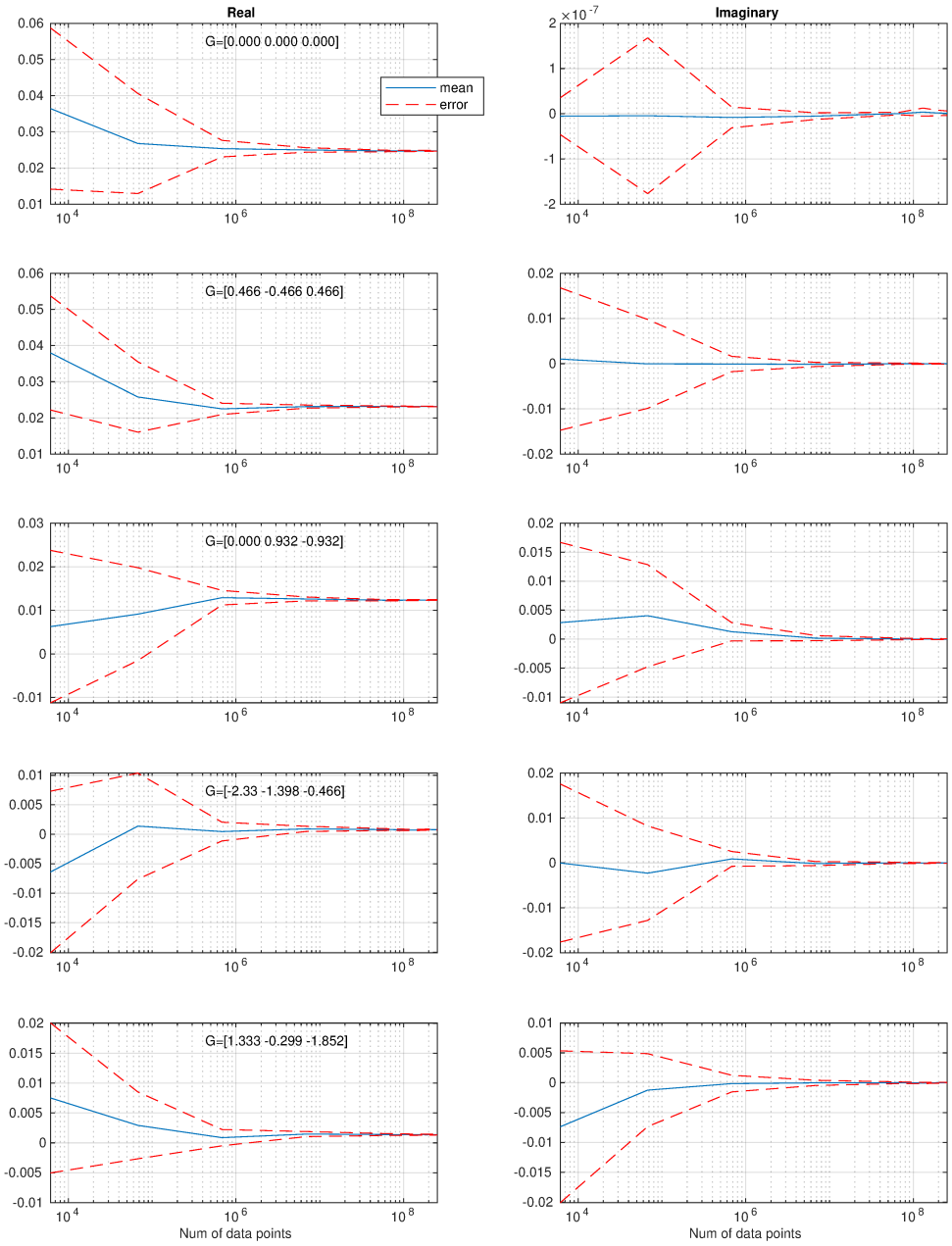

Figure B1: Test of convergence of APMD for different $\boldsymbol{G}$-points for diamond.

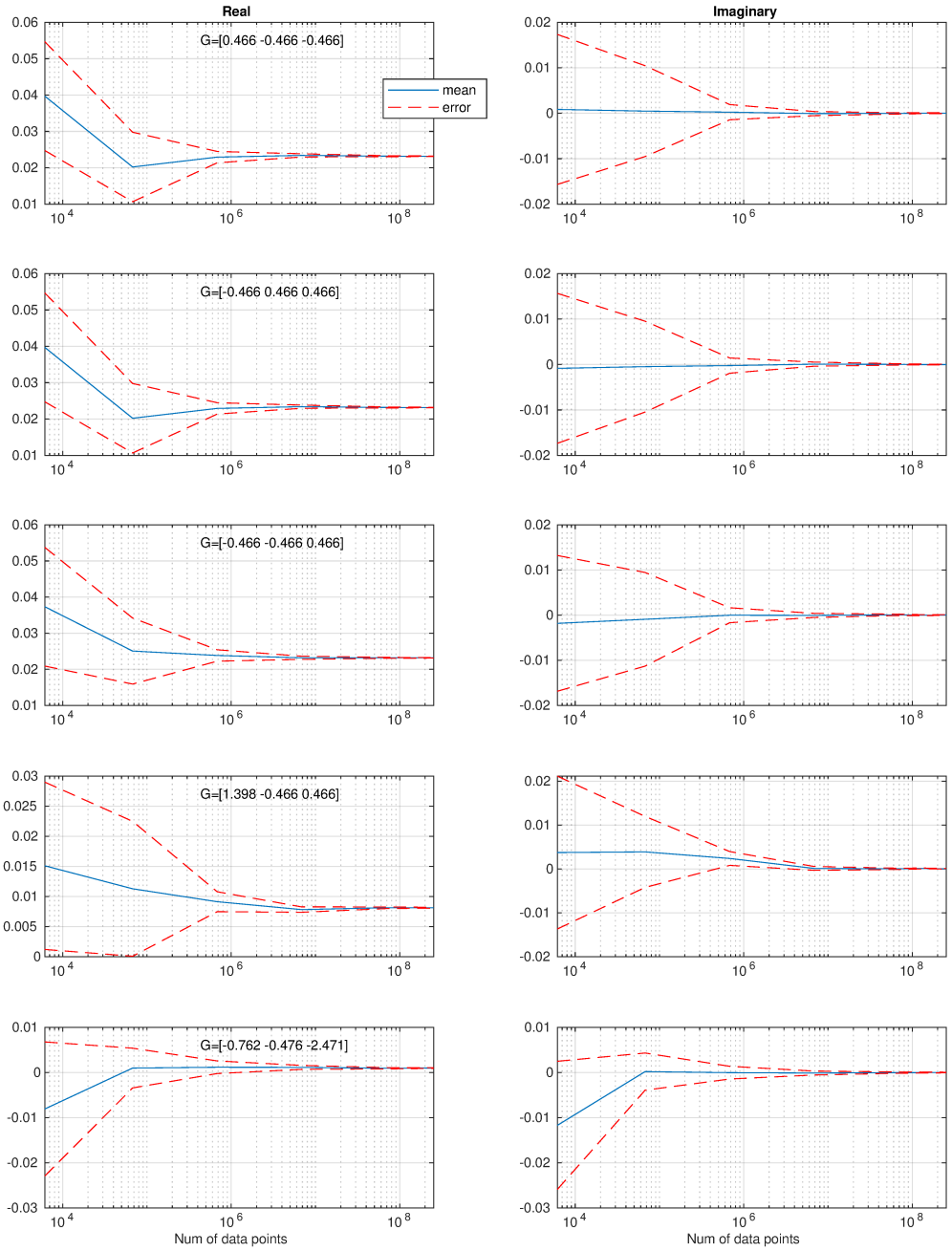

Figure B2: Test of convergence of APMD for different $\boldsymbol{G}$-points for diamond.

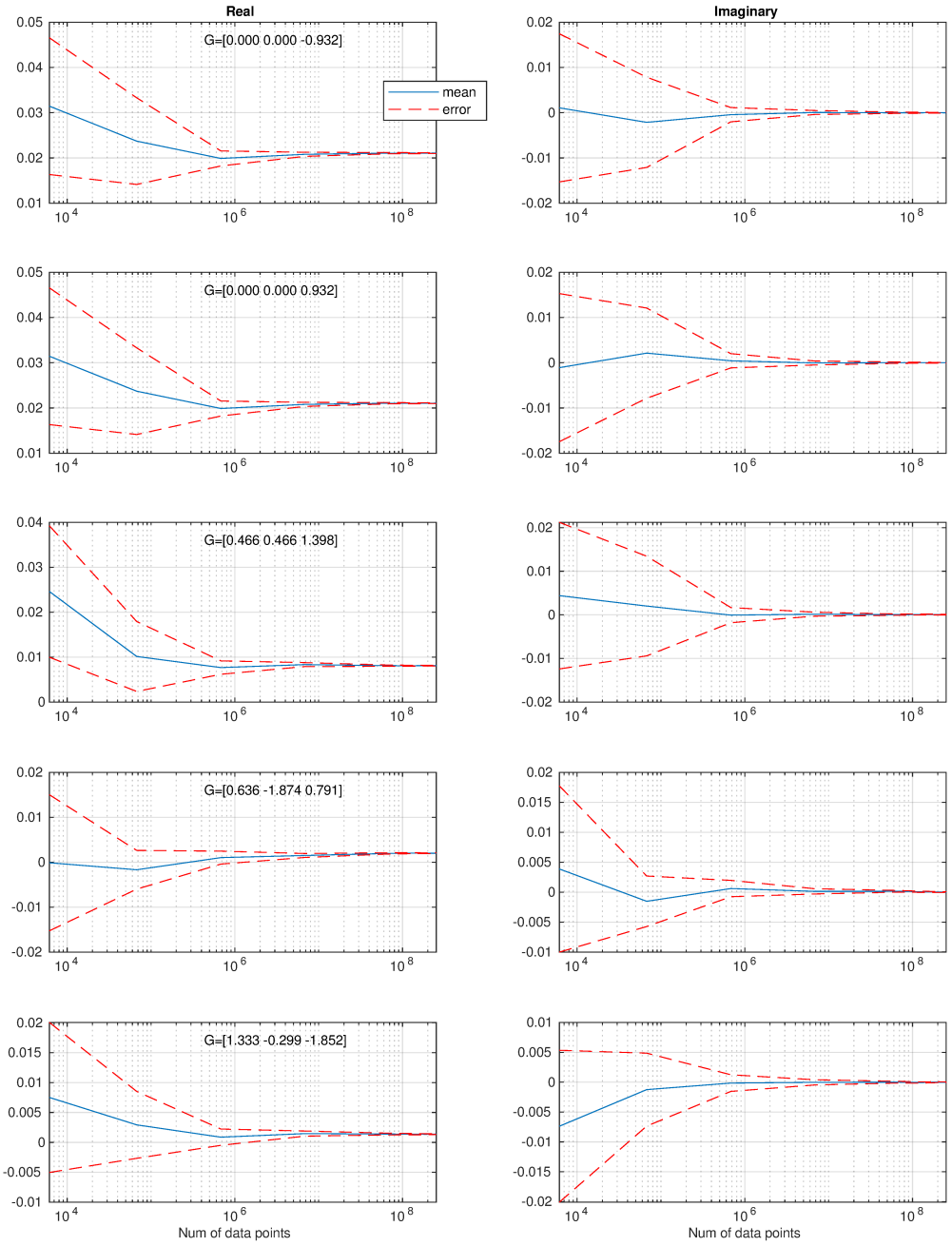

Figure B3: Test of convergence of APMD for different $\boldsymbol{G}$-points for diamond.

The following graphs show APMD values for silicon.

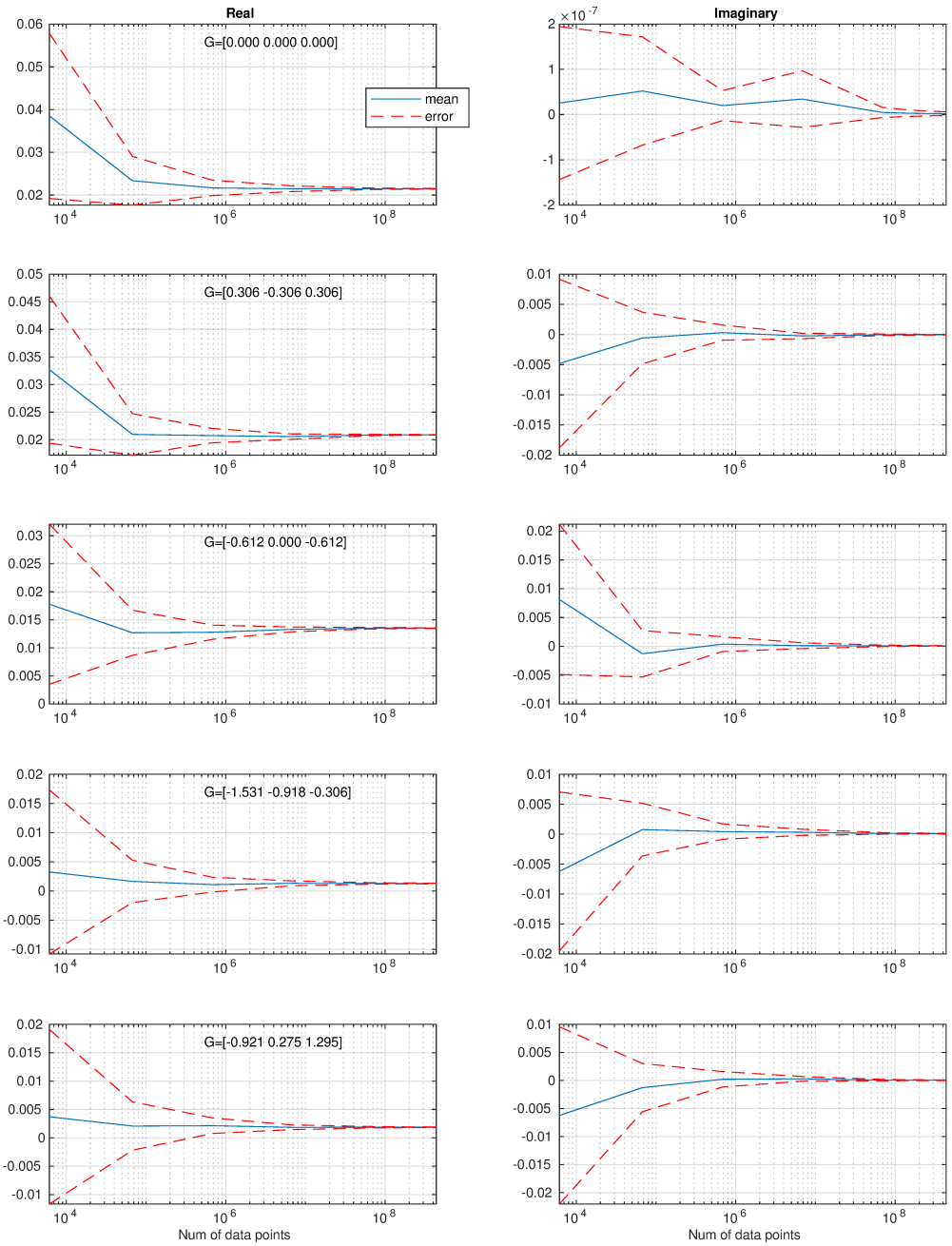

Figure B4: Test of convergence of APMD for different $\boldsymbol{G}$-points for silicon.

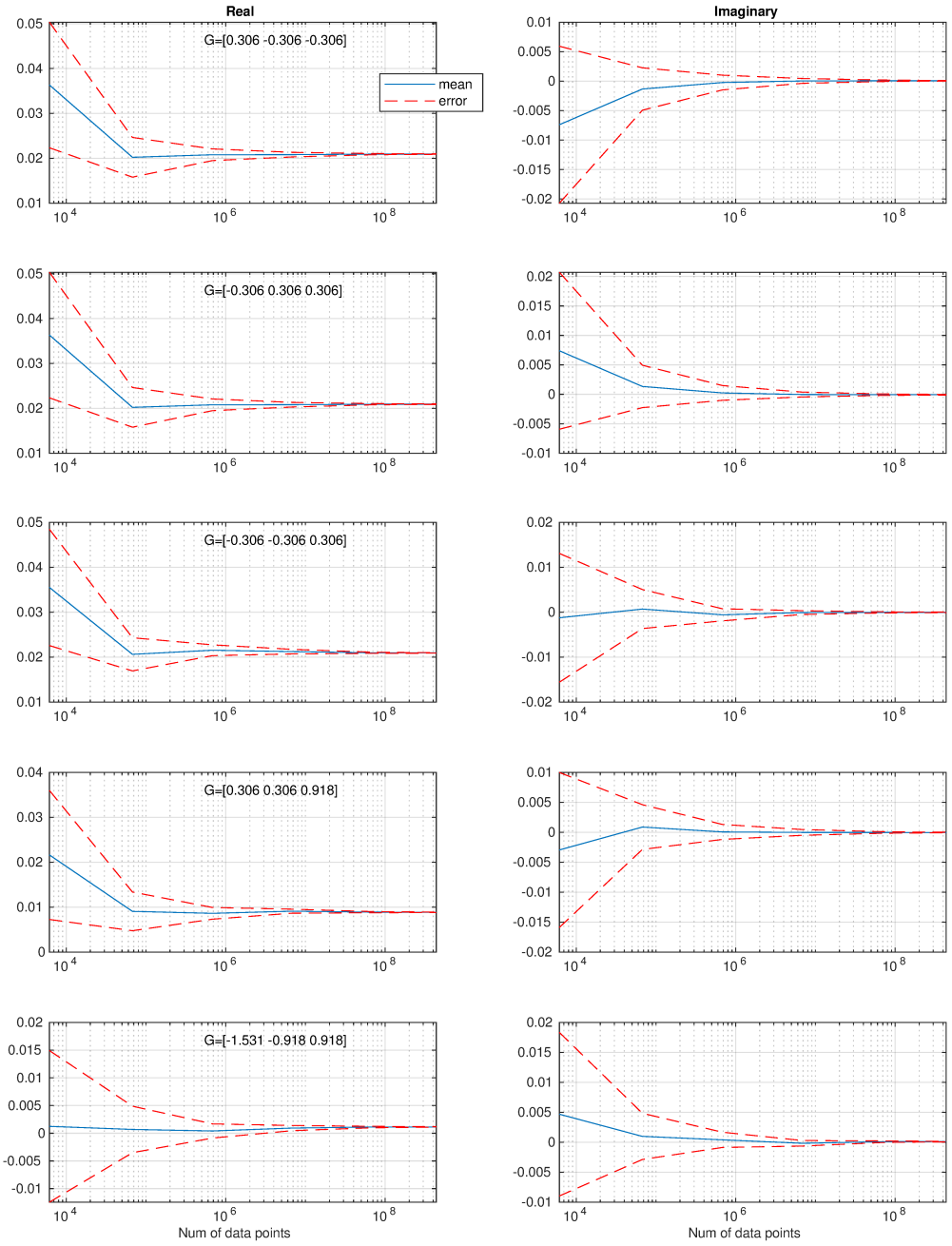

Figure B5: Test of convergence of APMD for different $\boldsymbol{G}$-points for silicon.

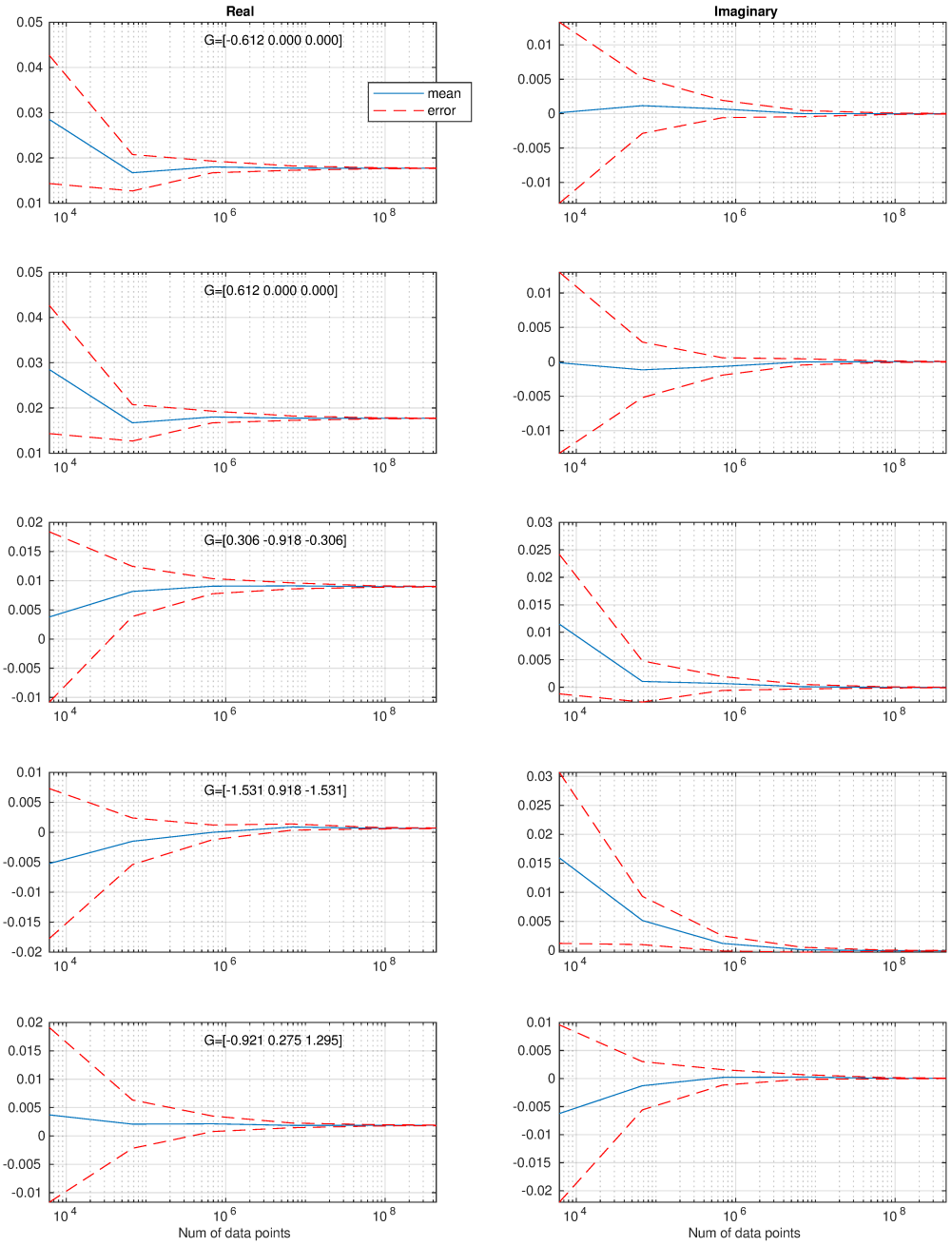

Figure B6: Test of convergence of APMD for different $\boldsymbol{G}$-points for silicon.

The following graphs show APMD values for germanium.

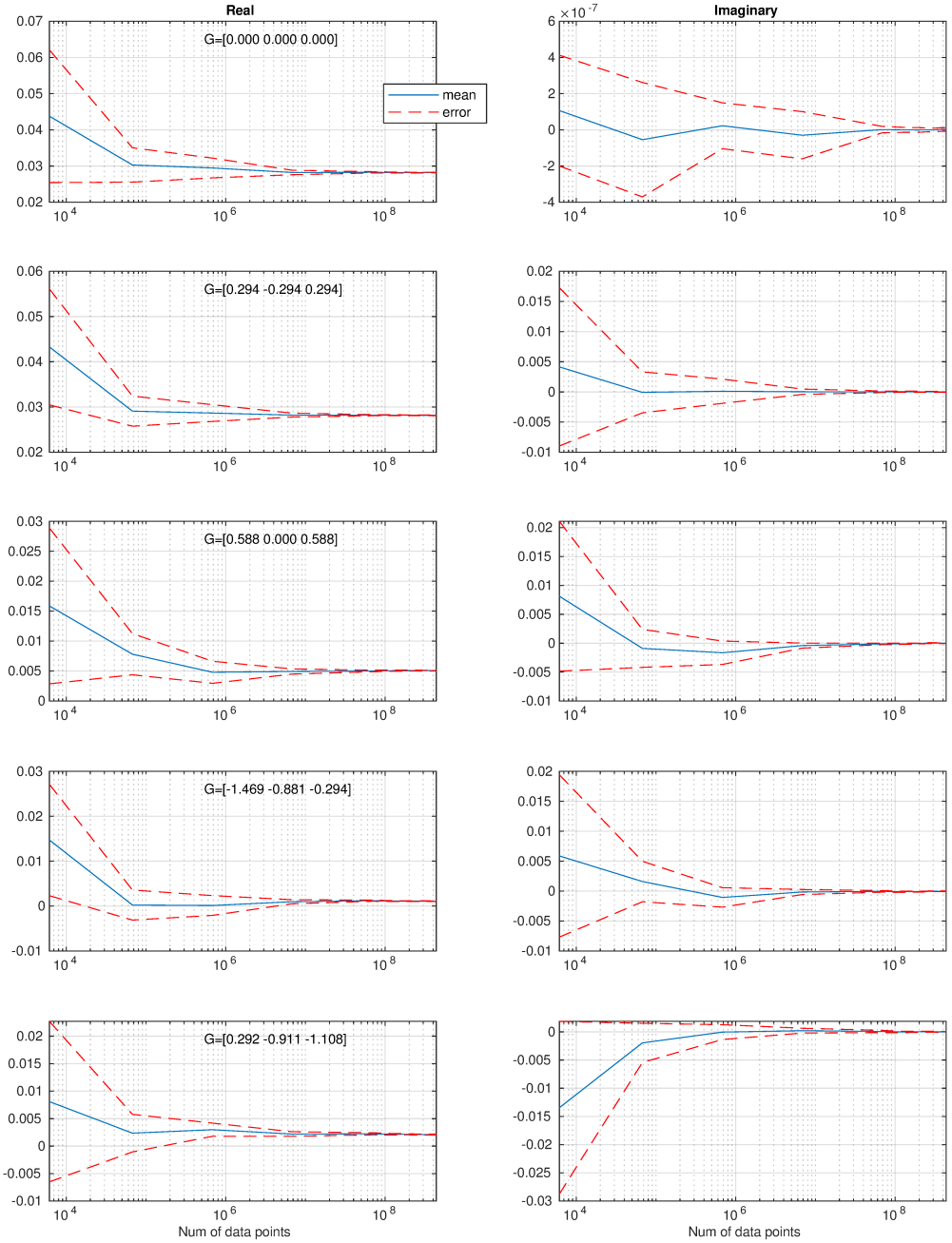

Figure B7: Test of convergence of APMD for different $\boldsymbol{G}$-points for germanium.

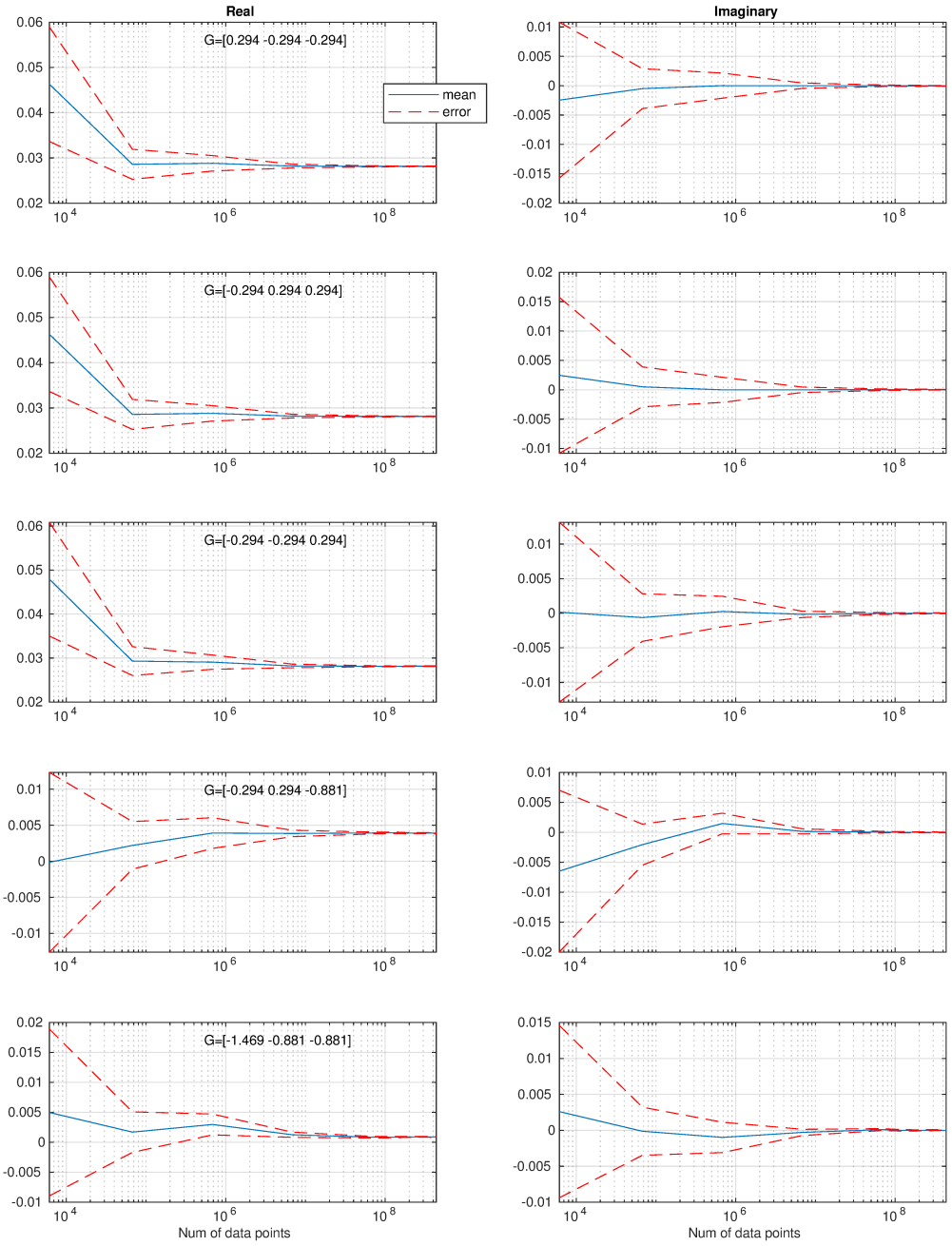

Figure B8: Test of convergence of APMD for different $\boldsymbol{G}$-points for germanium.

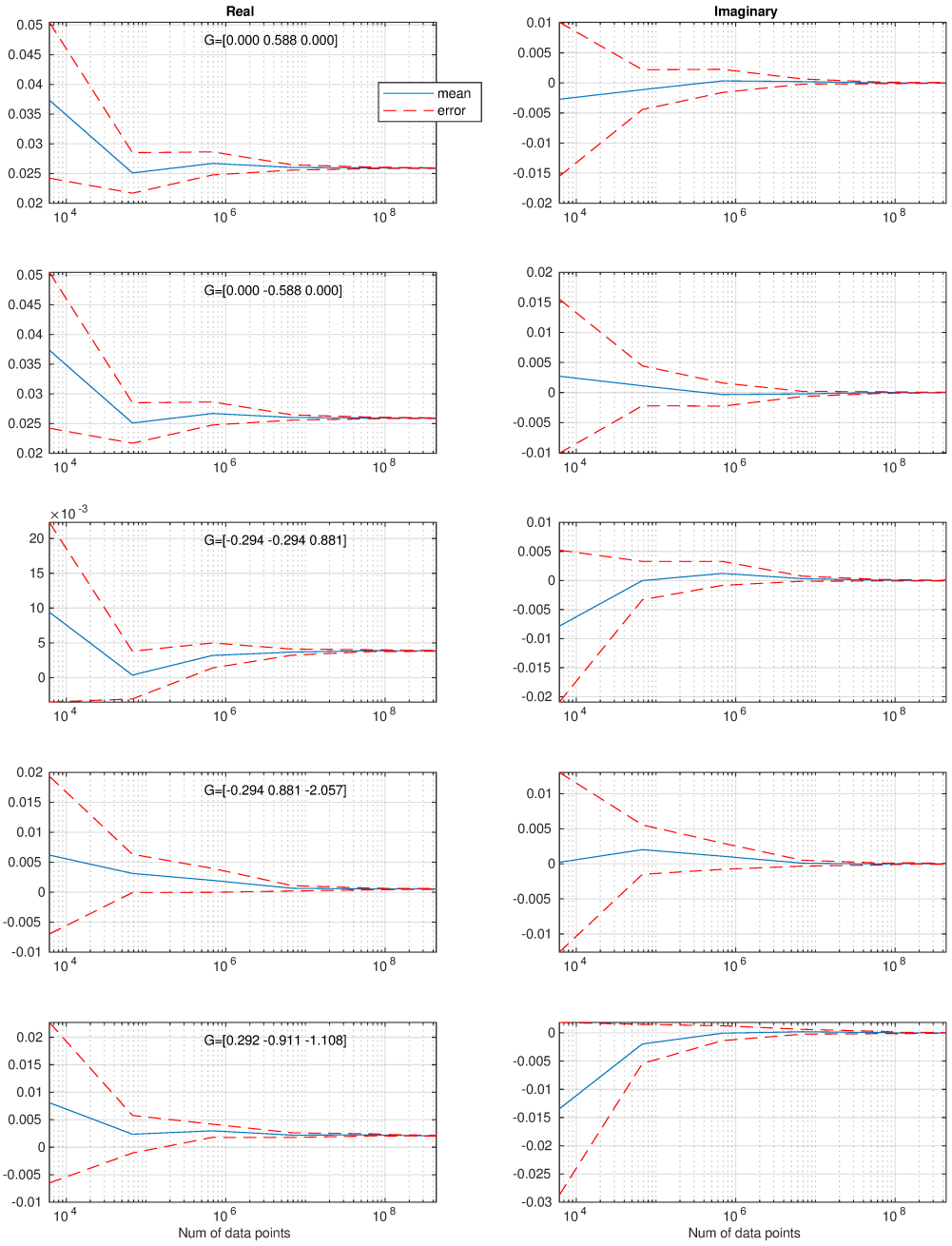

Figure B9: Test of convergence of APMD for different $\boldsymbol{G}$-points for germanium.